2nd

Optimal Space Communications Techniques

Final Report

January 22, 1973 - January 21, 1974

Goddard Space Flight Center

Greenbelt, Maryland

under

NASA Grant NGR 33-013-077

COMMUNICATIONS SYSTEMS LABORATORY

DEPARTMENT OF ELECTRICAL ENGINEERING

Donald L. Schilling

Professor of Electrical Engineering

Principal Investigator

THE CITY COLLEGE OF
THE CITY UNIVERSITY of NEW YORK

Optimal Space Communications Techniques

Final Report

January 22, 1973 - January 21, 1974

Goddard Space Flight Center

Greenbelt, Maryland

under

NASA Grant NGR 33-013-077

COMMUNICATIONS SYSTEMS LABORATORY
DEPARTMENT OF ELECTRICAL ENGINEERING

Donald L. Schilling
Professor of Electrical Engineering
Principal Investigator

Table of Contents

Introduction

This final eport summarizes one aspect of the research sponsored by the National Aeronautics and Space Administration under NASA Grant NGR-33-013-077 for the period January 22, 1973 through January 21, 1974. The research supported by this grant encompasses the problems concerned with the Digital Processing of Signals.

Part I of this report discusses digital multiplication of two waveforms using delta modulation. Delta modulation is an A/D conversion technique gaining widespread use. However, when a digital signal is to be processed it is first converted to a PCM signal and then processed using one's complement or two's complement arithmetic. We are currently studying DM-PCM and PCM-DM converters in which the DM employed is adaptive. Our results of this study should be completed by 1975.

It is important to note that the DM signal need not be converted to PCM prior to processing. The DM encoded signal can be processed digitally, directly in the DM format. The process of multiplication of two waveforms, each DM encoded, is discussed in this section. It is shown that while conventional multiplication of two N bit words requires N^2 complexity, multiplication using DM requires complexity which increases linearly with N.

Bounds on the SNR resulting from this multiplication are determined and compared with the SNR obtained using standard multiplication techniques.

A New Area of PLL research is discussed in Part II of this report. The PLL is one of the most often used systems in a communications system. It consists of a Phase Detector, Voltage Controlled Oscillator and a linear loop filter. The design of this filter is extremely important as the extra degrees of freedom given to the system by this filter results in the possibility of threshold extension increased pull-in range, decreased pull-in time, etc.

We are currently investigating PLL operation when the linear filter is replaced by a nonlinear digital processor. In particular we are considering a processor which "looks" at the sign of the past phase errors and corrects the VCO accordingly. The algorithm employed is similar to the one used in the DM and we are able to show an equivalence between the PLL and the DM.

The research supported by this grant and its "sister" grant NGR 33-013-063 has resulted in 1 PhD Dissertation (Part II) and the publication and presentation of several papers (Part III).

Participating in this program were:

Drs. J. Greco, I. Paz, W. Rosenberg and D. L. Schilling,

and

Messrs. J. LoCicero, D. Ucci, M. Steckman and N. Scheinberg.

I. Digital Processing of Signals: Digital Multiplication Using Delta Modulation

Abstract

We shall demonstrate that it is possible to multiply two signals that are digitally encoded with a Delta Modulator (DM) by operating on the incoming bit streams and employing only the standard digital manipulations. The resulting product error is investigated using signal statistics and for the case of constant inputs. Signal-to-quantization noise ratios (SNR) are obtained for constant inputs and various types of DM's.

1) Traditional Digital Multiplication

Generally digital multiplication is treated as a static operation; that is, two n-bit PCM words are either fed into a combinatoric circuit or into a preset read only memory (ROM) or a preset random access memory (RAM). The result is a 2 n-bit PCM word which is the product. If we wish to multiply two signals, both bandlimited to f_m, then we must perform this static operation on the PCM words from each signal every $\frac{1}{4f_m}$ seconds. This is because the product will be bandlimited to $2f_m$ as we know from the convolution $\leftrightarrow$ multiplication theorem of Fourier analysis.

The major objection to PCM multiplication is first, that it starts off fresh at each product operation and second, because of the combinatorics involved, the complexity involved and therefore the cost increases as the square of the numbers of bits that are used. We shall see that the DM multiplier not only operates dynamically with each product output depending on past inputs but its complexity varies linearly with the number of bits utlilized.

2) Digital Multiplication With Linear DM

A linear DM, whose block diagram is shown in Figure 1, produces a signal estimate

$$\hat{x}(i) = s \sum_{k=0}^{i-1} e_x(k), \tag{1}$$

where $s \triangleq$ DM step size.

Here we have assumed that x(t) is bandlimited to f_m and is sampled for DM encoding at $f_s >> f_m$.

Consider a second signal, y(t), which is also bandlimited to f_m and which is also passed thru a linear DM. The estimate of y(t) is

$$\hat{y}(i) = s \sum_{j=0}^{i-1} e_y(j). \tag{2}$$

We shall now form the product of the estimates and use this as an estimate of the product, that is,

$$\widehat{xy}(i) \approx \hat{x}(i)\,\hat{y}(i). \tag{3}$$

We can also write

$$\hat{x}(i-1) = s \sum_{k=0}^{i-2} e_x(k) \tag{4}$$

and

$$\hat{y}(i-1) = s \sum_{j=0}^{i-2} e_x(j). \tag{5}$$

Combining equations (1), (2), (4), and (5) we form a recursive relationship:

$$\hat{x}(i)\,\hat{y}(i) - \hat{x}(i-1)\,\hat{y}(i-1) = s \sum_{k=0}^{i-1} e_x(k) \cdot s \sum_{j=0}^{i-1} e_y(j) - s \sum_{k=0}^{i-2} e_x(k) \cdot s \sum_{j=0}^{i-2} e_y(j). \tag{6}$$

After some algebraic manipulations equation (6) reduces to

$$\hat{x}(i)\,\hat{y}(i) = \hat{x}(i-1)\,\hat{y}(i-1) + s^2\, e_y(i-1) \sum_{k=0}^{i-1} e_x(k) + s^2\, e_x(i-1) \sum_{j=0}^{i-1} e_y(j) - s^2\, e_x(i-1)\, e_y(i-1). \tag{7}$$

If we now introduce equation (3) into equation (7), the desired recursive relationship for the product is obtained:

$$\widehat{xy}(i) = \widehat{xy}(i-1) + s^2\, e_y(i-1) \sum_{k=0}^{i-1} e_x(k) + s^2\, e_x(i-1) \sum_{j=0}^{i-1} e_y(j) - s^2\, e_x(i-1)\, e_y(i-1). \tag{8}$$

The block diagram realization of equation (8) is shown in Figure 2. It is important to realize that in Figure 2 we use only scalers (s or s^2), accumulators, delays, adders, and multiplication by e (k). Since e (k) is only +1 or -1, multiplication by e (k) is merely an exclusive-or operation. Thus we are using only the standard digital manipulations in this realization. We also see that this realization is independent of the number of bits used in the original DM. That is, if we increase the number of bits, we do not have to add new operations to this realization, but merely increase the capacity of the adders, delays, etc.

3) Statistical Error Analysis

For any DM system the error produced is the difference between the actual input signal and the DM estimate signal after it has been filtered to eliminate the high frequency components introduced by the DM itself. Let us represent this error as:

$$\varepsilon_x = x - \hat{x}. \tag{9}$$

If we assume that x and $\hat{x}$ are jointly Gaussian, have zero means, and are highly correlated (as they should be because $\hat{x}$ is meant to be a good replica of x), then the joint probability density function is given as:

$$p_{x,\hat{x}}(\alpha, \beta) = \frac{1}{2\pi\, \sigma_x\, \sigma_{\hat{x}} \sqrt{1-\rho^2_{x\hat{x}}}} \exp\left\{-\left[\frac{1}{2(1-\rho^2_{x\hat{x}})}\right]\cdot\right. \tag{10}$$

$$\left.\left[\frac{\alpha^2}{\sigma_x^2} - \frac{2\,\rho_{x\hat{x}}\,\alpha\,\beta}{\sigma_x\,\sigma_{\hat{x}}} + \frac{\beta^2}{\sigma^2_{\hat{x}}}\right]\right\}$$

where

$\sigma_x^2 = E(x^2)$ = Variance of x,

$\sigma_{\hat{x}}^2 = E(\hat{x}^2)$ = Variance of $\hat{x}$,

$\rho_{x\hat{x}} = \frac{E(x\hat{x})}{\sigma_x\,\sigma_{\hat{x}}}$ = correlation cofficient.

Now we can apply the transformation $(x, \hat{x}) \rightarrow (x, \epsilon_x)$ given by

$$\epsilon_x = x - \hat{x}$$

in order to obtain the joint probability density function $p_{x,\epsilon_x}(\alpha, \beta)$, which shall be used later to determine the product error. This type of transformation is well known [1], and the joint probability density function is also jointly Gaussian. It is given as:

$$p_{x,\epsilon_x}(\alpha,\beta) = \frac{1}{2\pi\,\sigma_x\,\sigma_{\epsilon_x}\sqrt{1-\rho^2_{x\epsilon_x}}}\ \exp\left\{-\left[\frac{1}{2\,(1-\rho^2_{x\epsilon_x})}\right]\cdot\right.$$

$$\left.\left[\frac{\alpha^2}{\sigma_x^2} - \frac{2\,\rho_{x\epsilon_x}}{\sigma_x\,\sigma_{\epsilon_x}}\,\alpha\,\beta + \frac{\beta^2}{\sigma_{\epsilon_x}^2}\right]\right\} \tag{11}$$

where

$$\sigma_{\epsilon_x}^2 = E(\epsilon_x^2) = \sigma_x^2 - 2\,\rho_{x\hat{x}}\,\sigma_x\,\sigma_{\hat{x}} + \sigma_{\hat{x}}^2,$$

$$\rho_{x\epsilon_x} = \frac{\sigma_x^2 - \rho_{x\hat{x}}\,\sigma_x\,\sigma_{\hat{x}}}{\sigma_x\,\sigma_{\epsilon_x}} = \frac{\overline{x\epsilon_x}}{\sigma_x\,\sigma_{\epsilon_x}}.$$

In a completely analogous manner, we can assume y and $\hat{y}$ are jointly Gaussian, have zero means and are highly correlated and express $p_{y,\hat{y}}(\alpha,\beta)$ just like equation (10) except with σ_y, $\sigma_{\hat{y}}$ and $\rho_{y\hat{y}}$ replacing σ_x, $\sigma_{\hat{x}}$, and $\rho_{x\hat{x}}$ respectively. Then we can apply the transformation

$$\epsilon_y = y - \hat{y} \tag{12}$$

and obtain $p_{y\epsilon_y}(\alpha,\beta)$ exactly like equation (11), again with σ_y, σ_{ϵ_y}, and $\rho_{y\epsilon_y}$ replacing σ_x, σ_{ϵ_x}, and $\rho_{x\epsilon_x}$ respectively.

Now we can form the product error:

$$\epsilon_p = xy - \widehat{xy} \approx xy - \hat{x}\,\hat{y}. \tag{13}$$

Using equations (9) and (12) in (13) we obtain

$$\epsilon_p = xy - (x - \epsilon_x)(y - \epsilon_y),$$

or

$$\epsilon_p = x\,\epsilon_y + y\,\epsilon_x - \epsilon_x\,\epsilon_y. \tag{13a}$$

Under the assumption that x and y are statistically independent,

$$\bar{\epsilon}_p = 0 , \tag{14}$$

and the mean-square error is the same as the variance; that is,

$$\begin{aligned} \text{Var}(\epsilon_p) = \overline{\epsilon_p^2} = \overline{x^2}\ \overline{\epsilon_y^2} + \overline{y^2}\ \overline{\epsilon_x^2} + 2\,\overline{x\epsilon_x}\ \overline{y\,\epsilon_y} - 2\,\overline{x\epsilon_x}\ \overline{\epsilon_y^2} \\ - 2\,\overline{y\,\epsilon_y}\ \overline{\epsilon_x^2} + \overline{\epsilon_x^2}\ \overline{\epsilon_y^2} . \end{aligned} \tag{15}$$

All of the terms above are well defined from the above assumed and transformed statistics. If we substitute the variances and correlation coefficients for x, $\hat{x}$, y, and $\hat{y}$ into equation (15), it becomes (see Appendix A):

$$\overline{\epsilon_p^2} = \sigma_x^2 \sigma_y^2 + \sigma_{\hat{x}}^2 \sigma_{\hat{y}}^2 - 2\,\rho_{x\hat{x}}\,\rho_{y\hat{y}}\,\sigma_x\,\sigma_{\hat{x}}\,\sigma_y\,\sigma_{\hat{y}} . \tag{16}$$

Now if we make use of the assumptions that

$$\sigma_x \approx \sigma_{\hat{x}} ,$$

$$\sigma_y \approx \sigma_{\hat{y}} ,$$

and, since we use the same DM for x and y, we can assume that

$$\rho_{x\hat{x}} = \rho_{y\hat{y}} \triangleq \rho ,$$

then equation (16) simplifies to

$$\overline{\epsilon_p^2} = 2\,\sigma_x^2\,\sigma_y^2\,(1-\rho^2) . \tag{17}$$

Equation (17) is plotted verses ρ in Figure 3. Here we see that the closer the correlation coefficient is to ± 1, the smaller the mean square error. Thus a measure of the quality of the product is the correlation coefficient between the signal and the estimate; i.e., how well the estimate approximates the original signal.

4) Digital Multiplication Using Adaptive DM

Realizing that the main disadvantage with linear DM is its fixed step size and therefore its lack of dynamic range, we would like to extend our operation of digital multiplication to signals that have been encoded with an adaptive DM. One type of adaptive DM is the enhanced Abate mode which is described, for an input signal x, by the following set of equations:

$$\hat{x}(k) = \hat{x}(k-1) + S_x(k), \tag{18}$$

where

$$S_x(k) = |S_x(k-1)|\, e_x(k-1) + S\, e_x(k-2), \tag{18a}$$

$$= \text{step size at the } k^{th} \text{ interval},$$

$$e_x(k) = \text{sgn}\,(x(k) - \hat{x}(k)), \tag{18b}$$

and S = magnitude of the minimum step size.

We can also express the estimate, $\hat{x}(k)$, non-recursively, as:

$$\hat{x}(k) = \sum_{i=0}^{k} S_x(i)\,. \tag{18c}$$

Likewise, a signal y applied to an enhanced Abate mode DM yields:

$$\hat{y}(k) = \hat{y}(k-1) + S_y(k)\,. \tag{19}$$

where

$$S_y(k) = |S_y(k-1)|\, e_y(k-1) + S\, e_y(k-2), \tag{19a}$$

$$e_y(k) = \operatorname{sgn}(y(k) - \hat{y}(k)), \tag{19b}$$

and also

$$\hat{y}(k) = \sum_{j=0}^{k} S_y(j). \tag{19c}$$

Now if we form the product estimate as the product of the signal estimates and apply equations (18) and (19), we obtain

$$\hat{xy}(k) \approx \hat{x}(k)\,\hat{y}(k) = (\hat{x}(k-1) + S_x(k)) \cdot (\hat{y}(k-1) + S_y(k)). \tag{20}$$

If we expand equation (20) and employ equations (18c) and (19c), the following recursive relationship is obtained:

$$\hat{xy}(k) = \hat{xy}(k-1) + S_y(k) \sum_{i=0}^{k-1} S_x(i) + S_x(k) \sum_{j=0}^{k-1} S_y(j) + S_x(k)\, S_y(k). \tag{21}$$

Equation (21) is not as easily realized as equation (8) was for the linear DM so we must expand the last three terms of it to obtain a usable realization.

Starting with the last term of equation (21) and using equations (18a) and (19a) with k-1 replacing k, the result is:

$$S_x(k)\, S_y(k) = |S_x(k-1)\, S_y(k-1)|\, e_x(k-1)\, e_y(k-1) + S[\, |S_x(k-1)|\, e_x(k-1)\, e_y(k-2) + |S_y(k-1)|\, e_y(k-1)\, e_x(k-2)\,] + S^2\, e_x(k-2)\, e_y(k-2)\,. \quad (22)$$

Equation (22) is a recursive relationship that is easily realizable since taking the magnitude of a quantity is a simple digital operation and the other operations are merely standard digital manipulations.

The second and third terms of equation (21) appear to be a bit more difficult than the last term, but they can both be realized by introducing one clever substitution. Expressing the second and third terms of equation (21) in $\hat{x}(k)$ and $\hat{y}(k)$ notation, we obtain:

$$S_y(k) \sum_{i=0}^{k-1} S_x(i) = S_y(k)\, \hat{x}(k-1)\,, \quad (23)$$

and

$$S_x(k) \sum_{j=0}^{k-1} S_y(j) = S_x(k)\, \hat{y}(k-1). \quad (24)$$

A recursive relationship can now be developed for $S_y(k)\, \hat{x}(k-1)$ by using equation (19a), equation (18) with k-1 replacing k, and the relationship that is always true for the step size of an enhanced Abate mode DM:

$$S_y(k-1) = |S_y(k-1)|\, e_y(k-2) \quad (25)$$

or

$$|S_y(k-1)| = S_y(k-1)\, e_y(k-2)\,. \quad (26)$$

After the above substitutions and some manipulation, we realize the second term of equation (21) as:

$$S_y(k)\,\hat{x}(k-1) = S_y(k-1)\,\hat{x}(k-2)\,e_y(k-1)\,e_y(k-2) + S_x(k-1)\,S_y(k-1)\,e_y(k-1)\,e_y(k-2) + \hat{x}(k-1)\,S\,e_y(k-2)\ . \tag{27}$$

Similarly a recursive relation for $S_x(k)\,\hat{y}(k-1)$ can be obtained by employing equation (18a), equation (19) with k-1 replacing k, and the step size relationship:

$$S_x(k-1) = |S_x(k-1)|\,e_x(k-2) \tag{28}$$

or

$$|S_x(k-1)| = S_x(k-1)\,e_x(k-2)\ . \tag{29}$$

The third term of equation (21) is thus realized as follows:

$$S_x(k)\,\hat{y}(k-1) = S_x(k-1)\,\hat{y}(k-2)\,e_x(k-1)\,e_x(k-2) + S_y(k-1)\,S_x(k-1)\,e_x(k-1)\,e_x(k-2) + \hat{y}(k-1)\,S\,e_x(k-2)\ . \tag{30}$$

Observing equation (27) and equation (30) we see that $S_x(k-1)\,S_y(k-1)\,e_y(k-1)\,e_y(k-2)$ and $S_y(k-1)\,S_x(k-1)\,e_x(k-1)\,e_x(k-2)$

are needed for these realization. However, we have already obtained $S_x(k)\, S_y(k)$ recursively via equation (22); so we merely need to delay $S_x(k)\, S_y(k)$ and then multiply it by $e_y(k-1)\, e_y(k-2)$ and $e_x(k-1)\, e_x(k-2)$ respectively. This then completes our realization. The block diagram for the entire adaptive digital multiplier is shown in Figures 4, 4a, 4a (i), 4a (ii), 4b and 4c.

5) Signal-To-Quantization Noise Ratio For Constant Inputs

Let us first consider the linear DM responding to an input x which is a constant. In order to obtain the quantization noise, we must realize that it is introduced via the A/D converter and therefore we must differentiate between the sampled value of x, $x(k)$, and the sampled and A/D converted (or quantized) value of x, $x_q(k)$. The input-output quantization relationship is shown in Figure 5. Thus

$$x_q(k) = mS \,,$$

where m = an integer,

and S = DM step size,

if $(mS - \frac{1}{2}S) < x(k) < (mS + \frac{1}{2}S)$.

In Figure 6 we show the response of a linear DM to a constant input. It is important to note that the arithmetic used in the digital DM is offset binary and one characteristic of offset binary is that the difference between two identical numbers is a "zero minus". This means that when $x_q(k) = \hat{x}(k)$, $e_x(k)$ will be -1, and we will always underestimate our signal. Thus the linear DM is a biased estimator when using offset binary arithmetic.

As we have done before, let us assume that the input x is Gaussian with probability density function:

$$p_x(\alpha) = \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left(-\alpha^2/2\sigma_x^2\right) . \tag{31}$$

If we examine $\hat{x}$ when it has reached steady state, we see that it is repetitive and takes on values $(x_q - s)$ and x_q as shown in regions R_1 and R_2 in Figure 6. Now if we also assume that S is sufficiently small, then the sampled input x (k) is equally likely to occur in the interval $(x_q - \frac{1}{2}S\ ,\ x_q + \frac{1}{2}S)$. Therefore, the steady state error

$$\epsilon_x(k) = x(k) - \hat{x}(k) \tag{32}$$

will take on a uniform probability density function $\left[p_{\epsilon_x}(\alpha) \right]$. To determine $p_{\epsilon_x}(\alpha)$ we use the law of total probability as applied to probability density functions:

$$p_{\epsilon_x}(\alpha) = p_{\epsilon_x}(\alpha \mid \hat{x} \in R_1)\, P(R_1) + p_{\epsilon_x}(\alpha \mid \hat{x} \in R_2)\, P(R_2) . \tag{33}$$

It is reasonable to conclude that since we are seeking the steady state error that regions R_1 and R_2 are equally likely when $\hat{x}$ has reached steady state. Thus

$$P(R_1) = P(R_2) = \tfrac{1}{2} . \tag{33a}$$

Returning to the conditional densities in equation (33) and the previous assumption that S is sufficiently small so that x (k) is equally

likely for the interval $((x_q - \frac{1}{2}S,\ x_q + \frac{1}{2}S)$ it is not difficult to see that:

$$p_{\epsilon_x}(\alpha \mid \hat{x} \in R_1) = \frac{1}{S} \quad , \qquad \tfrac{1}{2}S < \alpha < 3S/2 \ ,$$

$$= 0 \quad , \qquad \text{elsewhere,} \tag{33b}$$

and

$$p_{\epsilon_x}(\alpha \mid \hat{x} \in R_2) = \frac{1}{S} \quad , \qquad -\tfrac{1}{2}S < \alpha < \tfrac{1}{2}S \ ,$$

$$= 0 \quad , \qquad \text{elsewhere.} \tag{33c}$$

Combining equations (33) thru (33c), we obtain:

$$p_{\epsilon_x}(\alpha) = \frac{1}{2S} \quad , \qquad -\tfrac{1}{2}S < \alpha < 3S/2 \ ,$$

$$= 0 \quad , \qquad \text{elsewhere.} \tag{34}$$

Now we see that $p_{\epsilon_x}(\alpha)$ is uniformly distributed over the interval $(-\frac{1}{2}S\ ,\ 3S/2)$, and we can prove that the linear DM is a biased estimator by showing that $\overline{\epsilon_x} \neq 0$. Instead we can calculate

$$\overline{\epsilon_x} = \int_{-\infty}^{\infty} \alpha\, p_{\epsilon_x}(\alpha)\, d\alpha \ ,$$

$$\overline{\epsilon_x} = \tfrac{1}{2}S \ . \tag{35}$$

Likewise we can calculate the second moment of ϵ_x and the variance of ϵ_x :

$$\overline{\epsilon_x^2} = \int_{-\infty}^{\infty} \alpha^2\, p_{\epsilon_x}(\alpha)\, d\alpha \ ,$$

$$\overline{\epsilon_x^2} = 7S^2/12 \ . \tag{35a}$$

$$\text{Var}(\epsilon_x) = \overline{\epsilon_x^2} - \overline{\epsilon_x}^2 ,$$

$$\text{Var}(\epsilon_x) = S^2/3 . \qquad (35b)$$

Referring to equation (31), the input signal power is:

$$\overline{x^2} = \int_{-\infty}^{\infty} \alpha^2 \, p_x(\alpha) \, d\alpha ,$$

$$\overline{x^2} = \sigma_x^2 . \qquad (36)$$

Thus we can form a signal-to-quantization noise ratio for a constant input:

$$\text{SNR} \triangleq \frac{\overline{x^2}}{\text{Var}(\epsilon_x)} = \frac{3\sigma_x^2}{2} . \qquad (37)$$

Analogous to equation (13a), we can form the product error as:

$$\epsilon_p = x\epsilon_y + y\epsilon_x - \epsilon_x\epsilon_y , \qquad (38)$$

where ϵ_x and ϵ_y now have probability density functions which include the A/D quantization [i.e., equation (34)]. Assuming that x and y are statistically independent and $p_x(\alpha) = p_y(\alpha)$ = equation (31), we obtain:

$$\overline{\epsilon_p} = -\overline{\epsilon_x}\,\overline{\epsilon_y} = -S^2/4 , \qquad (39)$$

and

$$\overline{\epsilon_p^2} = \overline{x^2}\,\overline{\epsilon_y^2} + \overline{y^2}\,\overline{\epsilon_x^2} + \overline{2x\epsilon_x}\,\overline{y\epsilon_y}$$

$$-2\overline{x\epsilon_x}\,\overline{\epsilon_y^2} - \overline{2y\epsilon_y}\,\overline{\epsilon_x^2} + \overline{\epsilon_x^2}\,\overline{\epsilon_y^2} . \qquad (40)$$

To completely evaluate equation (40), we must know the joint probability density functions $p_{x,\epsilon_x}(\alpha,\beta)$ and $p_{y,\epsilon_y}(\alpha,\beta)$, so that $\overline{x\,\epsilon_y}$ and $\overline{y\,\epsilon_y}$ can be determined. However we already know that the marginal densities are Gaussian (x and y) and uniform (ϵ_x and ϵ_y). Also the marginal densities must satisfy

$$p_x(\alpha) = \int_{-\infty}^{\infty} p_{x,\epsilon_x}(\alpha,\beta)\, d\beta \qquad (41)$$

and

$$p_{\epsilon_x}(\beta) = \int_{-\infty}^{\infty} p_{x,\epsilon_x}(\alpha,\beta)\, d\alpha \,. \qquad (41a)$$

With the previously specified marginal densities we conclude that

$$p_{x,\epsilon_x}(\alpha,\beta) = \frac{1}{2S\sqrt{2\pi\,\sigma_x^2}} \exp(-\alpha^2/2\,\sigma_x^2)\,, \quad -\tfrac{1}{2}S < \beta < 3S/2 \quad \forall\, \alpha \,. \qquad (41b)$$

This means that

$$p_{x,\epsilon_x}(\alpha,\beta) = p_x(\alpha)\; p_{\epsilon_x}(\beta) \,, \qquad (41c)$$

or that x and ϵ_x are statistically independent. Of course the same argument applies for y and ϵ_y. Evaluating equation (40), we see that:

$$\overline{\epsilon_p^2} = \overline{x^2}\;\overline{\epsilon_y^2} + \overline{y^2}\;\overline{\epsilon_x^2} + \overline{\epsilon_x^2}\;\overline{\epsilon_y^2} \,, \qquad (40a)$$

$$\overline{\epsilon_p^2} = \frac{.7S^2\,\sigma_x^2}{12} + \frac{7S^2\,\sigma_y^2}{12} + \frac{49\,S^4}{144} \,. \qquad (40b)$$

Now we can evaluate the variance of ϵ_p and use this as our quantization product noise:

$$\text{Var}(\epsilon_p) = \overline{\epsilon_p^2} - \overline{\epsilon_p}^2 \quad ,$$

$$\text{Var}(\epsilon_p) = \frac{7S^2}{12}\left(\sigma_x^2 + \sigma_y^2\right) + \frac{5S^2}{18} \quad . \tag{40c}$$

Defining the product signal-to-quantization noise ratio as

$$\text{SNR}_p \triangleq \frac{\overline{(xy)^2}}{\text{Var}(\epsilon_p)} \quad , \tag{42}$$

we obtain:

$$\text{SNR}_p = \frac{36\,\sigma_x^2\,\sigma_y^2}{21S^2\left(\sigma_x^2 + \sigma_y^2\right) + 10S^4} \quad . \tag{42a}$$

As a final comment, it is to be noted that the above SNR_p reflects performance before the final low pass filter where out of band noise is eliminated. Currently, investigations are being made to determine the performance after filtering.

Now we will repeat the above for the case of an adaptive DM operating in the enhanced Abate mode [equations (18) - (18c)]. It has been shown by Song, Garodnick and Schilling [2] that in this mode of operation all steady state responses take on a "general error pattern". The general error pattern as shown in Figure 7, is the steady state response $\hat{x}$ to a constant input signal x, which has been quantized to x_q. In the general error pattern, n is a non-negative integer, or

$$n = 0, 1, 2, 3, \ldots N \quad .$$

Employing the same conditions and assumptions as in the linear DM analysis, we determine $p_{\epsilon_x}(\alpha)$ thusly:

$$P_{\epsilon_x}(\alpha \mid n = \gamma) = p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_1)\, P(R_1) + p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_2)\, P(R_2) + p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_3)\, P(R_3) + p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_4)\, P(R_4), \tag{43}$$

$$P(R_i) = \tfrac{1}{4}, \quad i = 1, 2, 3, 4. \tag{43a}$$

As we can see from Figure 7,

$$p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_1) = 1/S, \quad (\gamma + \tfrac{1}{2})S < \alpha < (\gamma + 3/2)S,$$
$$= 0, \quad \text{elsewhere}, \tag{43b}$$

$$p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_2) = 1/S, \quad \tfrac{1}{2}S < \alpha < 3S/2,$$
$$= 0, \quad \text{elsewhere}, \tag{43c}$$

$$p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_3) = 1/S, \quad -(\gamma + \tfrac{1}{2})S < \alpha < -(\gamma - \tfrac{1}{2})S$$
$$= 0, \quad \text{elsewhere}, \tag{43d}$$

$$p_{\epsilon_x}(\alpha \mid n = \gamma, \hat{x} \in R_4) = 1/S, \quad -\tfrac{1}{2}S < \alpha < \tfrac{1}{2}S$$
$$= 0, \quad \text{elsewhere}. \tag{43e}$$

Combining equations (43) to (43e), we obtain:

$$\begin{aligned} p_{\epsilon_x}(\alpha \mid n = \gamma) &= 1/4S \quad , \quad (\gamma + \tfrac{1}{2})S < \alpha < (\gamma + 3/2)S \quad , \\ &= 1/4S \quad , \quad \tfrac{1}{2}S < \alpha < 3S/2 \quad , \\ &= 1/4S \quad , \quad -(\gamma + \tfrac{1}{2})S < \alpha < -(\gamma - \tfrac{1}{2})S \quad , \\ &= 1/4S \quad , \quad -\tfrac{1}{2}S < \alpha < \tfrac{1}{2}S \quad , \\ &= 0 \quad , \quad \text{elsewhere} \quad . \end{aligned} \tag{43f}$$

We also notice that the probability density function of n, $p_n(\gamma)$, is discrete with values at $\gamma = 0, 1, 2, \ldots, N$ which we shall represent as $P_n(\gamma)$, where

$$\sum_{\gamma=0}^{N} P_n(\gamma) = 1 \quad . \tag{44}$$

To obtain the desired $p_{\epsilon_x}(\alpha)$, we employ

$$p_{\epsilon_x}(\alpha) = \sum_{\gamma=0}^{N} p_{\epsilon_x}(\alpha \mid n = \gamma)\, P_n(\gamma) \quad . \tag{45}$$

The result is as follows:

[due to R_1]:

$$\begin{aligned} p_{\epsilon_x}(\alpha) &= P_n(0)/4S \quad , \quad \tfrac{1}{2}S < \alpha < 3S/2 \quad , \\ &= P_n(1)/4S \quad , \quad 3S/2 < \alpha < 5S/2 \quad , \\ &= P_n(2)/4S \quad , \quad 5S/2 < \alpha < 7S/2 \quad , \\ &\vdots \\ &= P_n(N)/4S \quad , \quad (N + \tfrac{1}{2})S < \alpha < (N + 3/2)S \quad , \end{aligned}$$

[due to R_2]:

$$p_{\epsilon_x}(\alpha) = 1/4S \quad , \quad \tfrac{1}{2}S < \alpha < 3S/2 \quad ,$$

$$[\text{due to } R_3]: \quad p_{\epsilon_x}(\alpha) = P_n(N)/4S \ , \quad -(N+\tfrac{1}{2})S < \alpha < -(N-\tfrac{1}{2})S \ ,$$

$$= P_n(N-1)/4S \ , \quad -(N-\tfrac{1}{2})S < \alpha < -(N-3/2)S \ ,$$

$$\vdots$$

$$= P_n(1)/4S \ , \quad -3S/2 < \alpha < -\tfrac{1}{2}S \ ,$$

$$= P_n(0)/4S \ , \quad -\tfrac{1}{2}S < \alpha < \tfrac{1}{2}S \ ,$$

$$[\text{due to } R_4]: \quad p_{\epsilon_x}(\alpha) = 1/4S \ , \quad -\tfrac{1}{2}S < \alpha < \tfrac{1}{2}S \ . \qquad (46)$$

Equation (46) is shown graphically in Figure 8. We see that this DM also produces a biased estimate because

$$\overline{\epsilon_x} = \tfrac{1}{2}S \ , \qquad (46a)$$

which can easily be verified by observing that $p_{\epsilon_x}(\alpha)$, as seen in Figure 8, is symmetric about $\alpha = \frac{1}{2}S$. Using equation (46), we can also calculate $\overline{\epsilon_x^2}$ (see Appendix B) and then the variance of ϵ_x:

$$\overline{\epsilon_x^2} = \frac{S^2}{12}(6\overline{n^2} + 6\overline{n} + 7) \ , \qquad (46b)$$

$$\text{Var}(\epsilon_x) = \frac{S^2}{6} \cdot (3\overline{n^2} + 3\overline{n} + 2) \ . \qquad (46c)$$

Now we obtain a signal-to-quantization noise ratio for a constant input:

$$\text{SNR} \triangleq \frac{\overline{x^2}}{\text{Var}(\epsilon_x)} = \frac{6\sigma_x^2}{S^2(3\overline{n^2} + 3\overline{n} + 2)} \ , \qquad (46d)$$

If we proceed to apply the same analysis as was done in equations (38) thru (42), we can obtain a product signal-to-quantization noise ratio for a constant input with this type of adaptive DM. The statistics

that we use are:

$$\overline{\epsilon_x} = \overline{\epsilon_y} = \tfrac{1}{2}S \ , \qquad (47)$$

and

$$\overline{\epsilon_x^2} = \overline{\epsilon_y^2} = \frac{S^2}{12}(6\,\overline{n^2} + 6\,\overline{n} + 7) \ . \qquad (47a)$$

Applying equations (39) and (40a) :

$$\overline{\epsilon_p} = -\overline{\epsilon_x}\,\overline{\epsilon_y} = -S^2/4 \qquad (48)$$

and

$$\overline{\epsilon_p^2} = \overline{x^2}\ \overline{\epsilon_y^2} + \overline{y^2}\ \overline{\epsilon_x^2} + \overline{\epsilon_x^2}\ \overline{\epsilon_y^2} \ ,$$

$$\overline{\epsilon_p^2} = \frac{S^2}{12}(\sigma_x^2 + \sigma_y^2)(6\,\overline{n^2} + 6\,\overline{n} + 7) + \frac{S^4}{144}(6\,\overline{n^2} + 6\,\overline{n} + 7)^2 . \qquad (48a)$$

It is significant to recall at this point that in order to obtain equation (40a), x and ϵ_x had to be statistically independent. Although at first glance it appears that $p_{\epsilon_x}(\alpha)$ [equation (46)] is dependent upon x, $p_{\epsilon_x}(\alpha)$ really depends on $p_x(\alpha)$ and as long as $p_x(\alpha)$ is specified, then $p_{\epsilon_x}(\alpha)$ is independent of the actual value of x. Thus equation (40a) is applicable in this case and equation (48a) is valid.

The conclusion is to now obtain the variance of ϵ_p and the product signal-to-quantization noise ratio:

$$\mathrm{Var}(\epsilon_p) = \frac{S^2}{12}(\sigma_x^2 + \sigma_y^2)(6\,\overline{n^2} + 6\,\overline{n} + 7) + \frac{S^2}{144}\left[(6\,\overline{n^2} + 6\,\overline{n} + 7)^2 - 9\right] \qquad (48b)$$

$$SNR_p \triangleq \frac{\overline{(xy)^2}}{Var\,(\varepsilon_p)} \quad ,$$

$$SNR_p = \frac{576\ \sigma_x^2\ \sigma_y^2}{12S^2\,(\sigma_x^2 + \sigma_y^2)\,(6\,\overline{n^2} + 6\,\overline{n} + 7) \;+\; S^4\,[(6\,\overline{n^2} + 6\,\overline{n} + 7)^2 - 9]} \quad . \qquad (49)$$

As with the linear DM, investigations are currently being made to determine performance after filtering and comparison made with computer simulation results.

6) Conclusion

As can be seen from the previous five sections, many facets of the digital multiplication using delta modulation problem are being investigated. There are many more inquires that must be made. Among the most pressing are a computer simulation and testing of the adaptive DM multiplier, a determination of exactly how much real time storage is needed, and analysis of performance after final low pass filtering. When these items are completed, digital multiplication using delta modulation can become an effective operation in digital processing of signals.

APPENDIX

A. Derivation of Equation (16):

Starting with equation (15):

$$\overline{\epsilon_p^2} = \overline{x^2}\,\overline{\epsilon_y^2} + \overline{y^2}\,\overline{\epsilon_x^2} + 2\,\overline{x\epsilon_x}\;\overline{y\epsilon_y} - 2\,\overline{x\epsilon_x}\;\overline{\epsilon_y^2} - 2\,\overline{y\epsilon_y}\,\overline{\epsilon_x^2} + \overline{\epsilon_x^2}\,\overline{\epsilon_y^2} \quad ,$$

$$= \overline{x^2}\,\overline{\epsilon_y^2} + \overline{y^2}\,\overline{\epsilon_x^2} + 2\,\overline{x\epsilon_x}\,(\overline{y\epsilon_y} - \overline{\epsilon_y^2}) + \overline{\epsilon_x^2}\,(\overline{\epsilon_y^2} - 2\,\overline{y\epsilon_y}) \quad ,$$

if we use $\overline{\epsilon_x^2}$ and $\overline{x\epsilon_x}$ from equation (11) and the analogous forms for $\overline{\epsilon_y^2}$ and $\overline{y\epsilon_y}$, the result is:

$$\overline{\epsilon_p^2} = \sigma_x^2(\sigma_y^2 - 2\rho_{y\hat{y}}\sigma_y\sigma_{\hat{y}} + \sigma_{\hat{y}}^2) + \sigma_y^2(\sigma_x^2 - 2\rho_{x\hat{x}}\sigma_x\sigma_{\hat{x}} + \sigma_{\hat{x}}^2)$$

$$+ 2(\sigma_x^2 - \rho_{x\hat{x}}\sigma_x\sigma_{\hat{x}})(\rho_{y\hat{y}}\sigma_y\sigma_{\hat{y}} - \sigma_{\hat{y}}^2)$$

$$+ (\sigma_x^2 - 2\rho_{x\hat{x}}\sigma_x\sigma_{\hat{x}} + \sigma_{\hat{x}}^2)(\sigma_{\hat{y}}^2 - \sigma_y^2) \quad .$$

If we now expand the above and cancel common terms, the final form becomes equation (16):

$$\overline{\epsilon_p^2} = \sigma_x^2\sigma_y^2 + \sigma_{\hat{x}}^2\sigma_{\hat{y}}^2 - 2\rho_{x\hat{x}}\rho_{y\hat{y}}\sigma_x\sigma_{\hat{x}}\sigma_y\sigma_{\hat{y}} \quad .$$

B. Derivation of Equation (46b):

Since $p_{\epsilon_x}(\alpha)$ is uniform for each of 2N intervals, if we let P_I represent the value in interval I, then the contribution to $\overline{\epsilon_x^2}$ for interval I is given as:

$$(\overline{\epsilon_x^2})_I = \int_I \alpha^2 P_I \, d\alpha = P_I \, \alpha^3/3 \Big|_I .$$

The total $\overline{\epsilon_x^2}$ is obtained by summing over all intervals.

$$\overline{\epsilon_x^2} = \sum_{i=0}^{N} \left\{ (P_n(i)/4S)(\alpha^3/3) \Big|_{(i+\frac{1}{2})S}^{(i+3/2)S} \right\}$$

$$+ \sum_{i=0}^{N} \left\{ (P_n(i)/4S)(\alpha^3/3) \Big|_{-(i+\frac{1}{2})S}^{-(i-\frac{1}{2})S} \right\}$$

$$+ (1/4S)(\alpha^3/3) \Big|_{-\frac{1}{2}S}^{3S/2} .$$

B. Derivation of Equation (46b): - continued

Inserting the limits and canceling terms, we obtain:

$$\overline{\epsilon_x^2} = \sum_{i=0}^{N} \left\{ \left(P_n(i)S^2/12 \right) \left[(i + 3/2)^3 - (i - \tfrac{1}{2})^3 \right] \right\} + 7S^2/24 \ ,$$

$$= \sum_{i=0}^{N} \left\{ \left(P_n(i)S^2/12 \right) (6\, i^2 + 6\, i + 7/2) \right\} + 7S^2/24 \ ,$$

$$= \tfrac{1}{2}S^2 \sum_{i=0}^{N} i^2 P_n(i) + \tfrac{1}{2}S^2 \sum_{i=0}^{N} i\, P_n(i) + (7S^2/24) \sum_{i=0}^{N} P_n(i) + 7S^2/24 \ ,$$

$$= \tfrac{1}{2}S^2\ \overline{n^2} + \tfrac{1}{2}S^2\ \overline{n} + 7S^2/12 \ .$$

Factoring the common terms yields equation (46b):

$$\overline{\epsilon_x^2} = \frac{S^2}{12} \left(6\, \overline{n^2} + 6\, \overline{n} + 7 \right) \ .$$

REFERENCES

[1] J.M. Wozencraft and I.M. Jacobs, Principles of Communication Engineering. N.Y.: John Wiley and Sons, Inc., 1965, pp. 153-154.

[2] C.L. Song, J. Garodnick and D.L. Schilling, "A Variable-Step-Size Robust Delta Modulator," IEEE Trans. Communications Technology, vol COM-19, pp. 1033-1044, Dec. 1971.

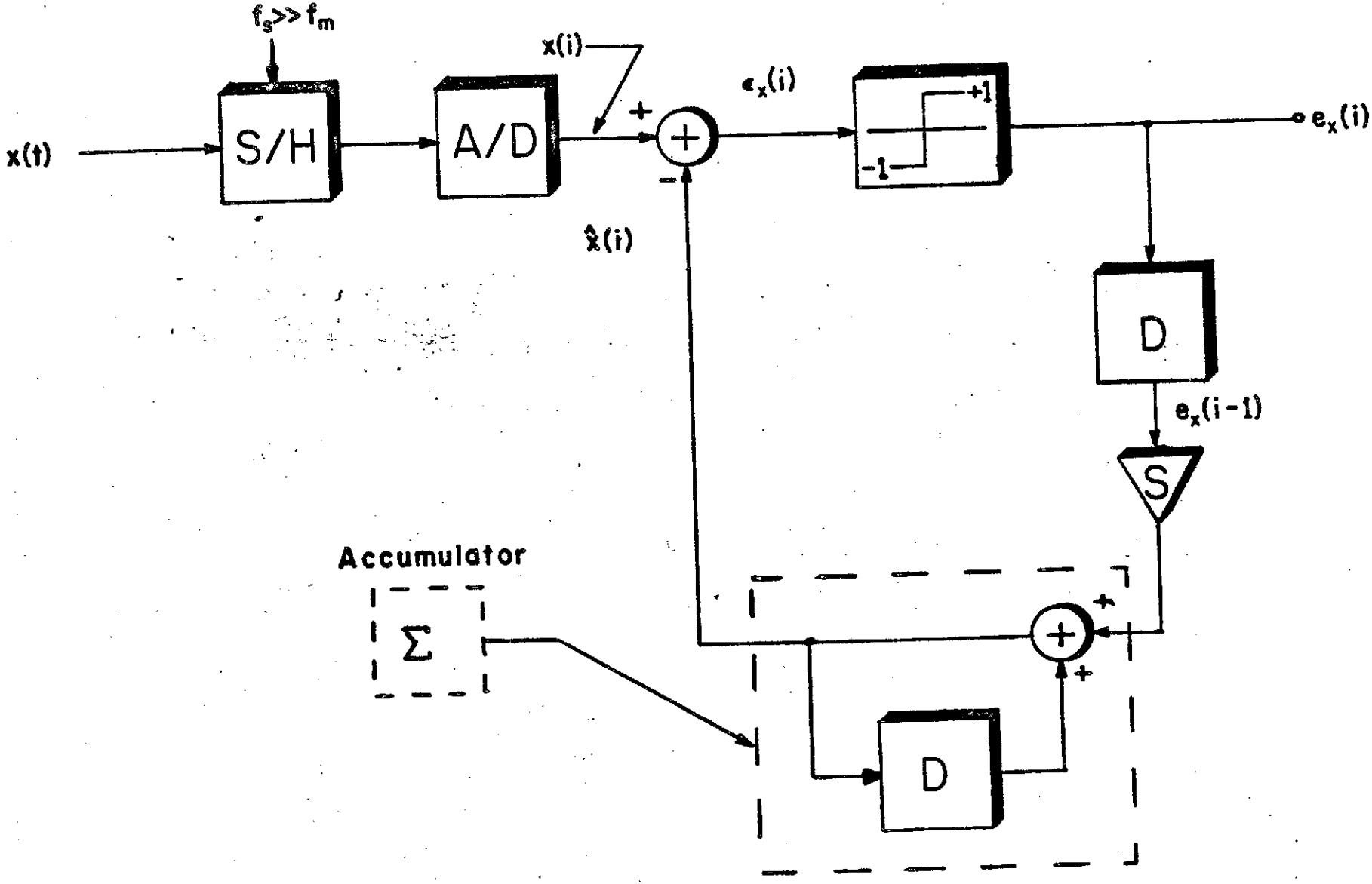


Figure 1. Linear Delta Modulator

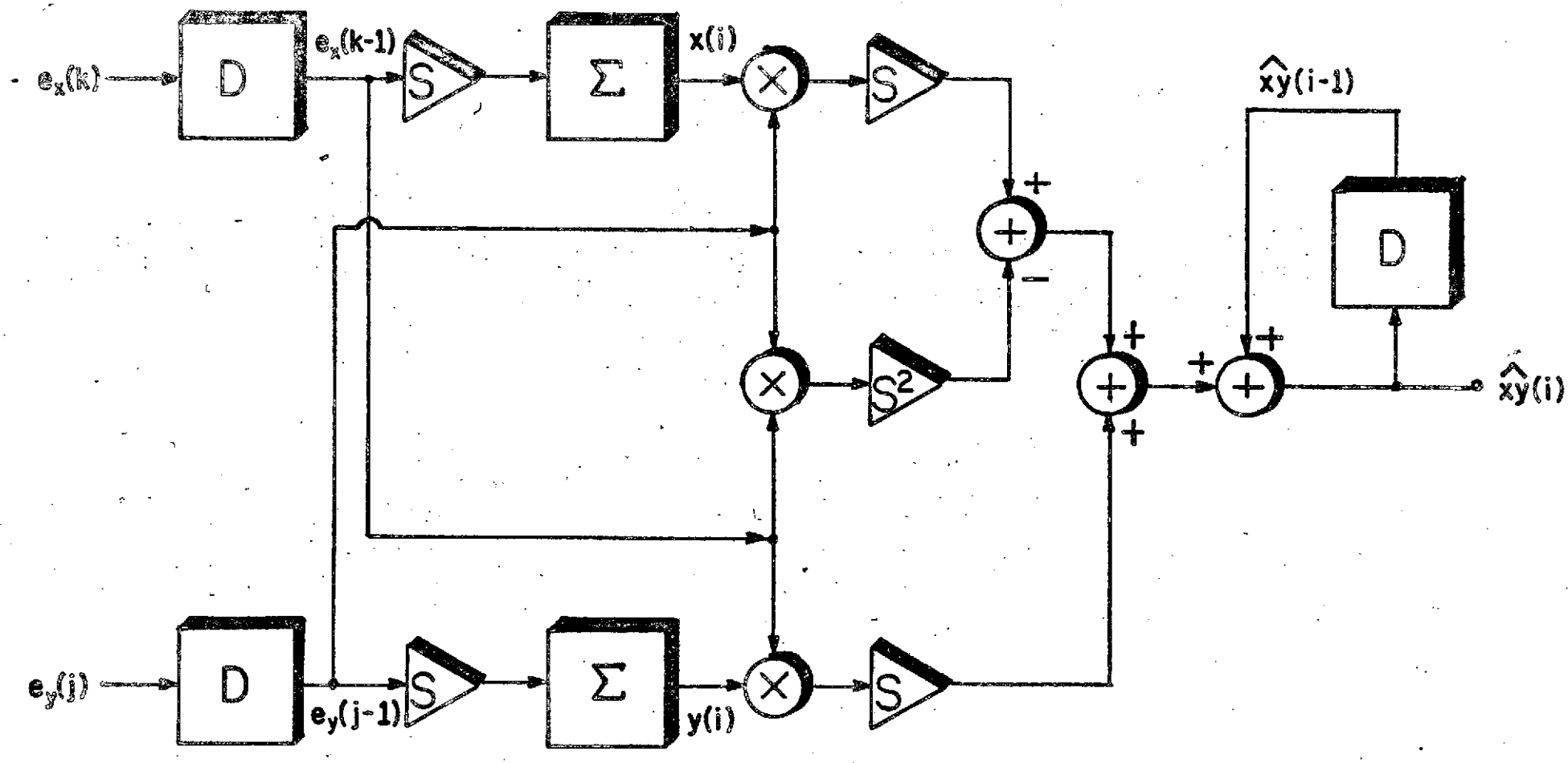


Figure 2. Digital Multiplier for Linear DM

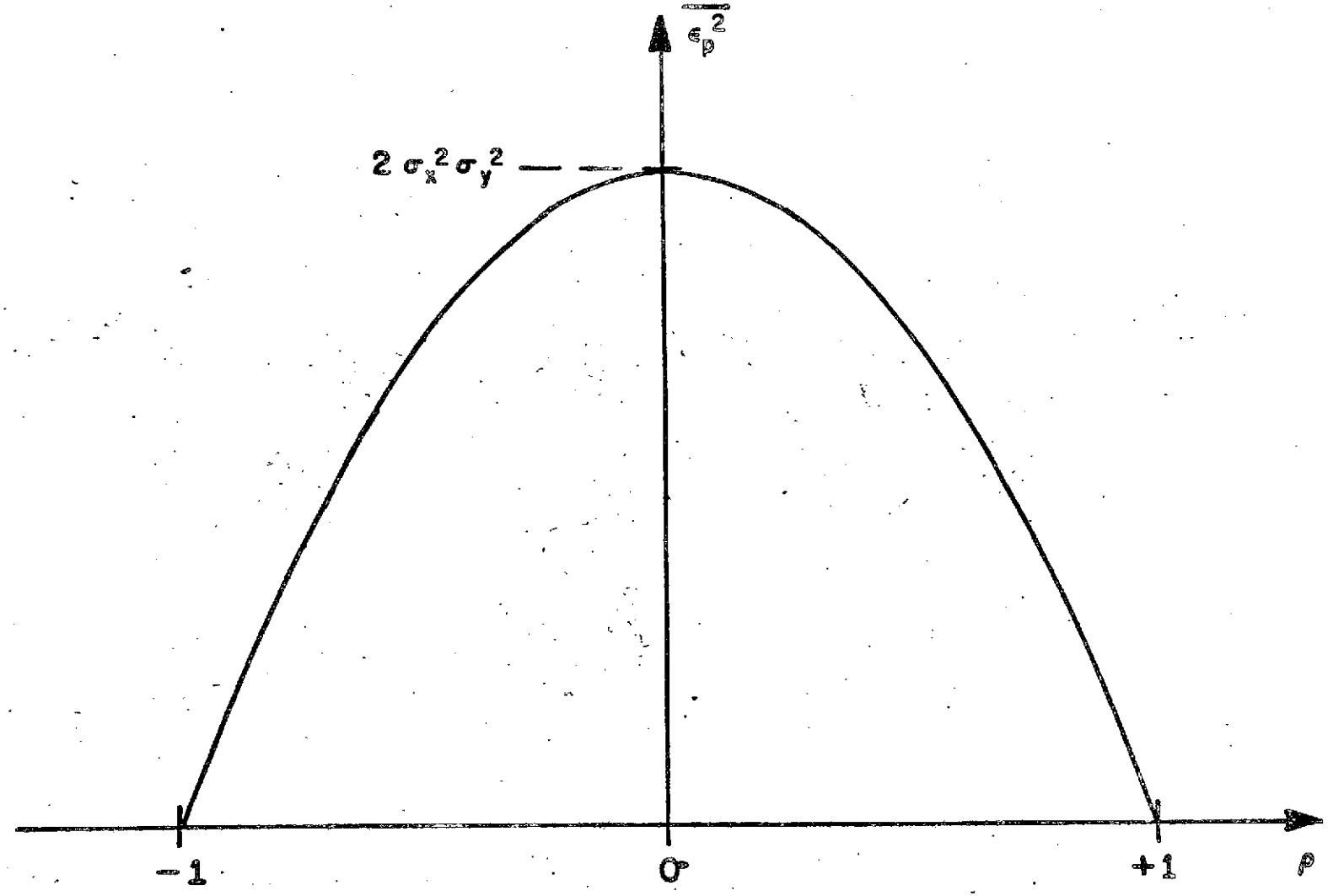


Figure 3. Product Mean Square Error vs. Correlation Coeffecient between Signal and Estimate

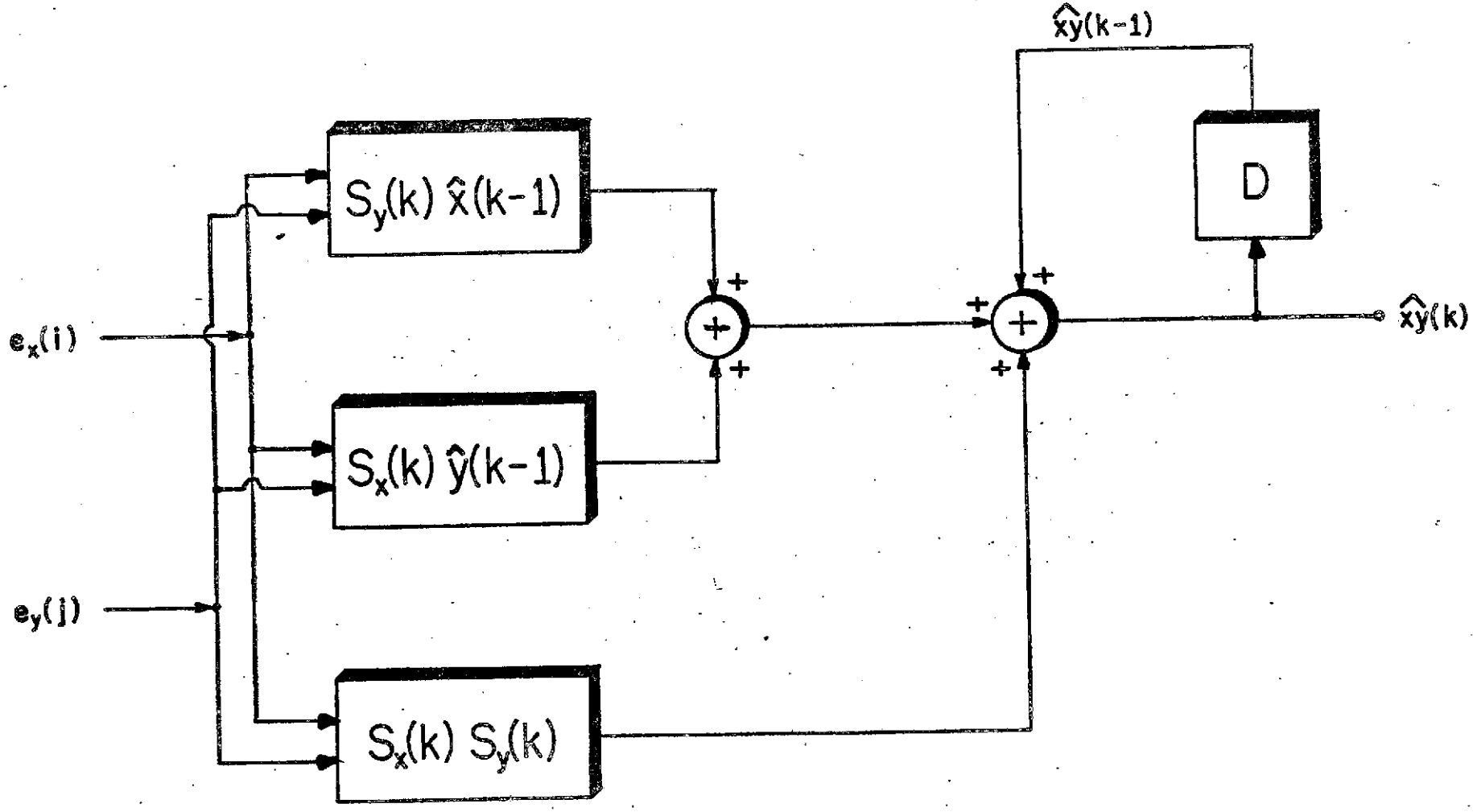


Figure 4. Digital Multiplier for Adaptive DM

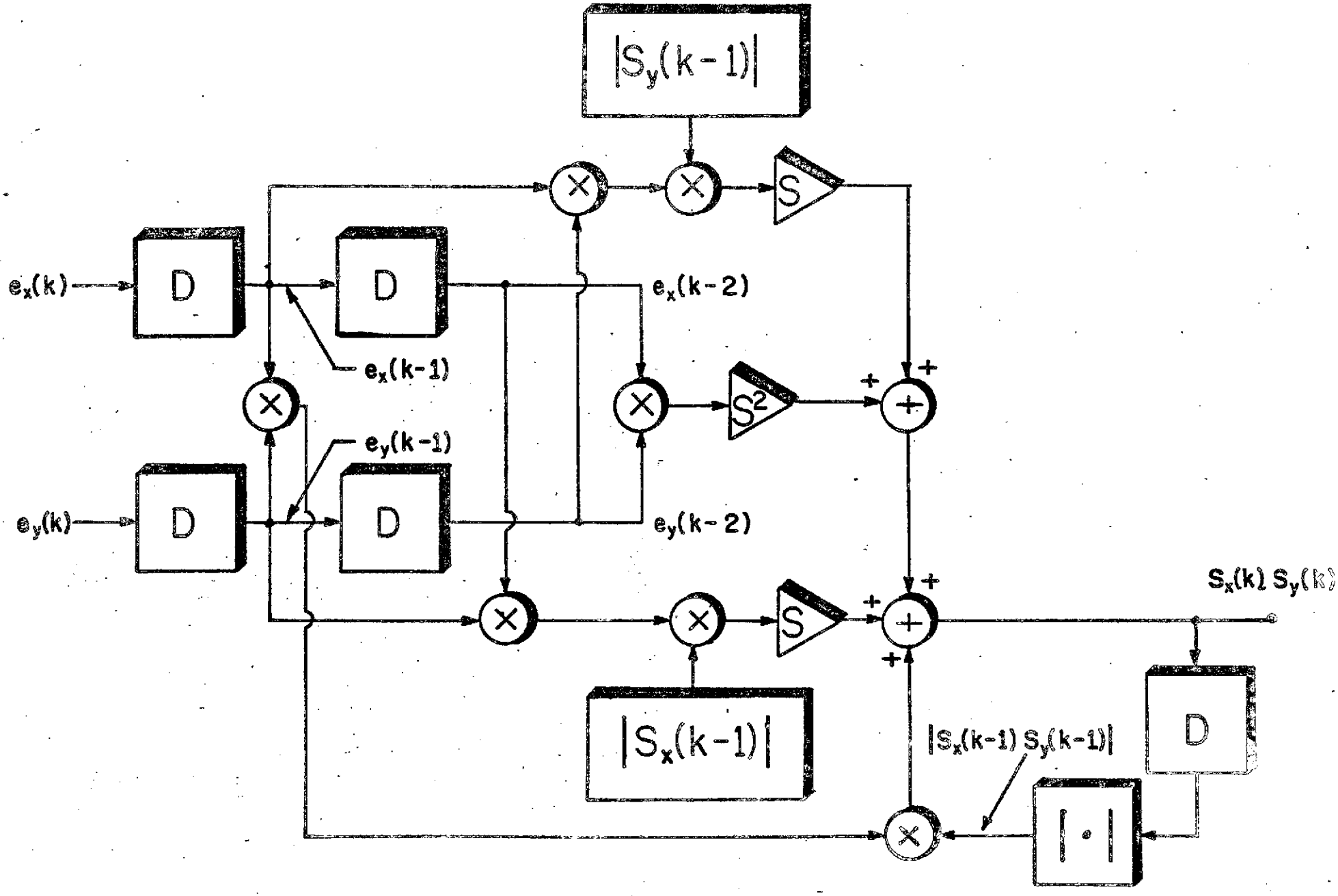


Figure 4a. $S_x(k)S_y(k)$ Realization for Enhanced Abate Mode Adaptive DM

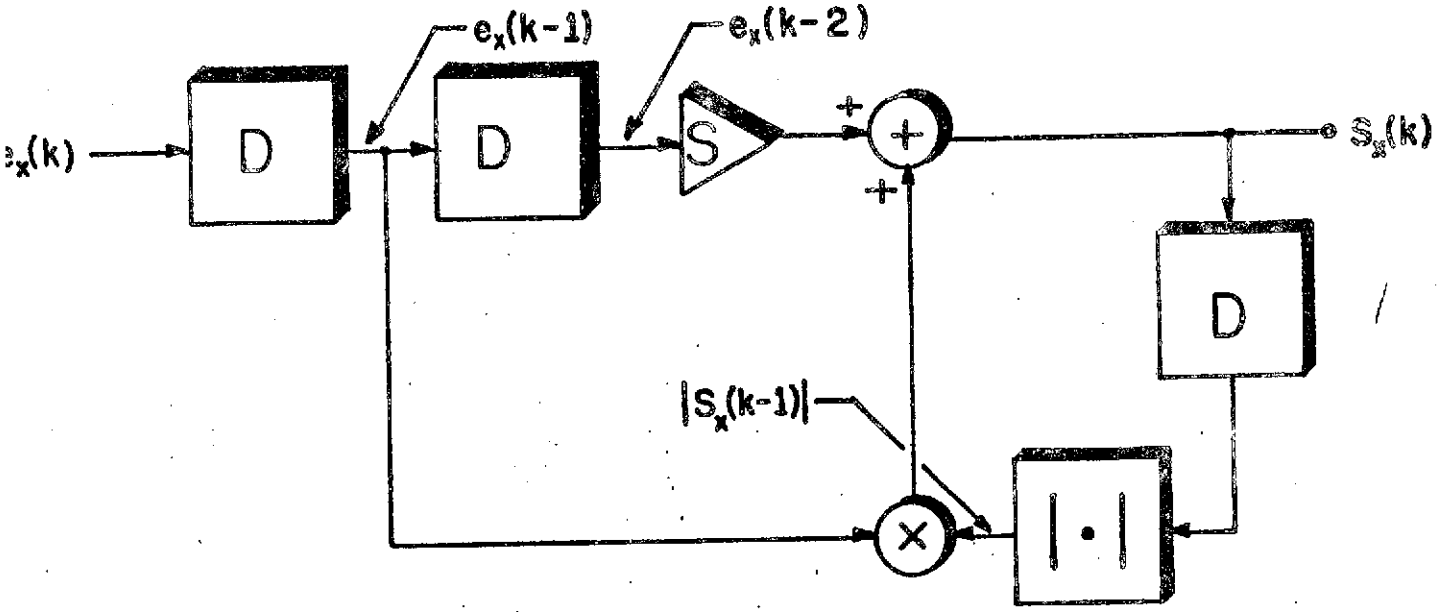


Figure 4a(i). Realization of $|S_x(k-1)|$

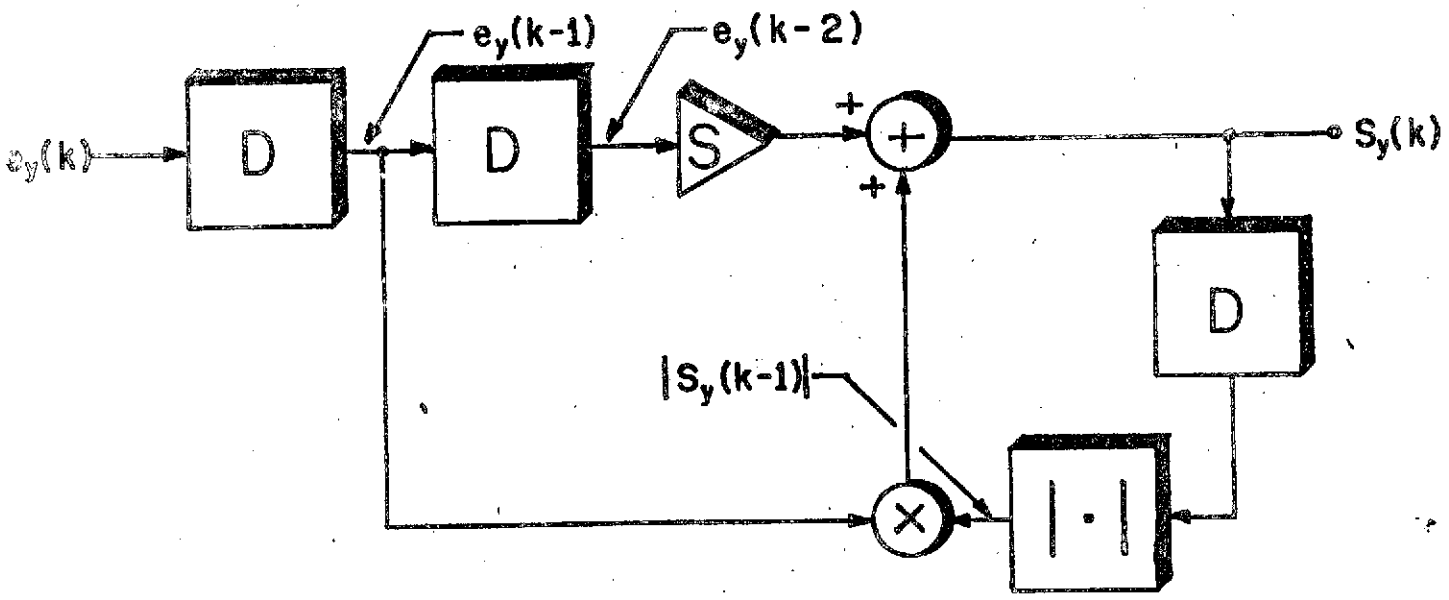


Figure 4a(ii). Realization of $|S_y(k-1)|$

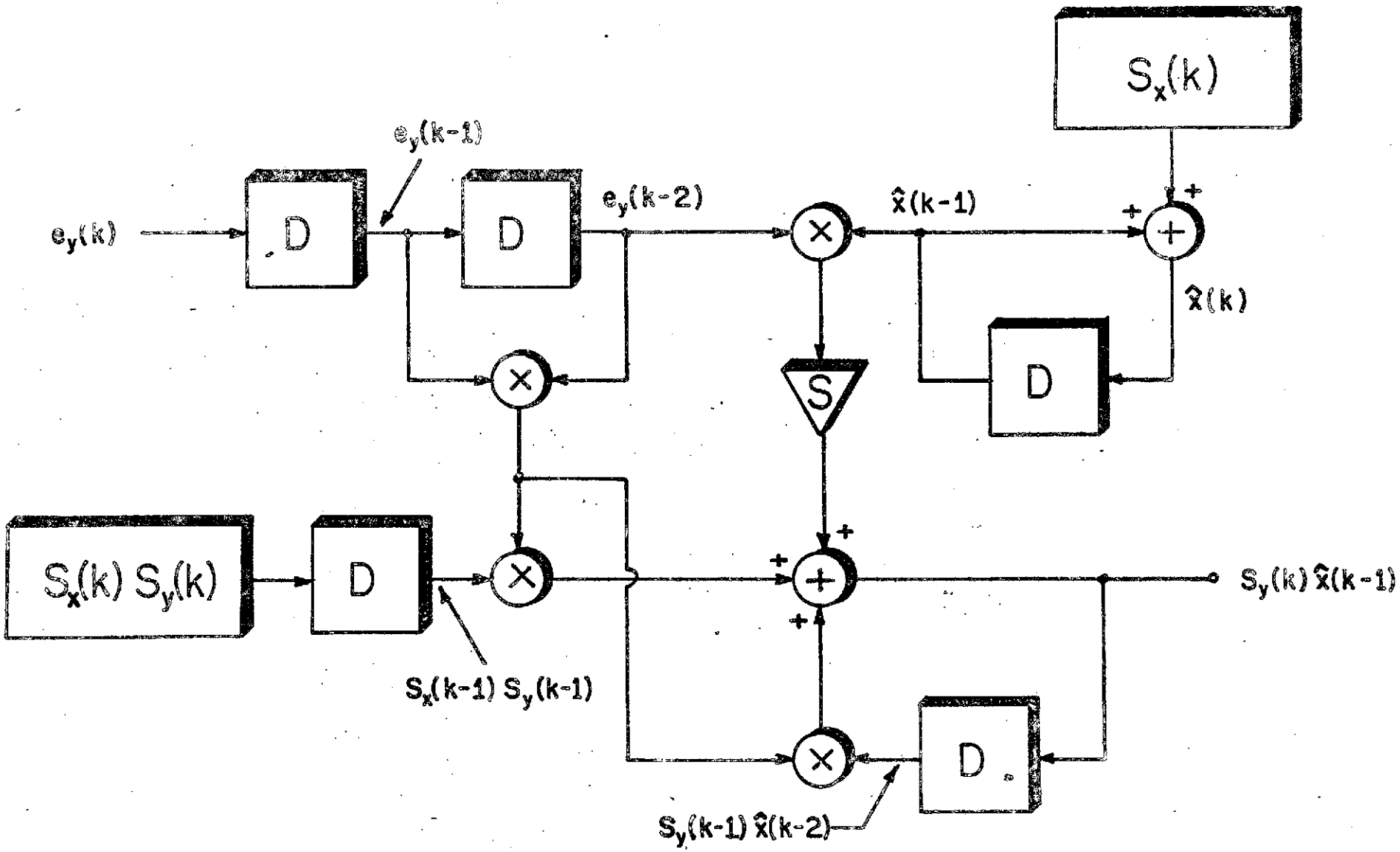


Figure 4b. Realization of $S_y(k)\,\hat{x}(k-1)$

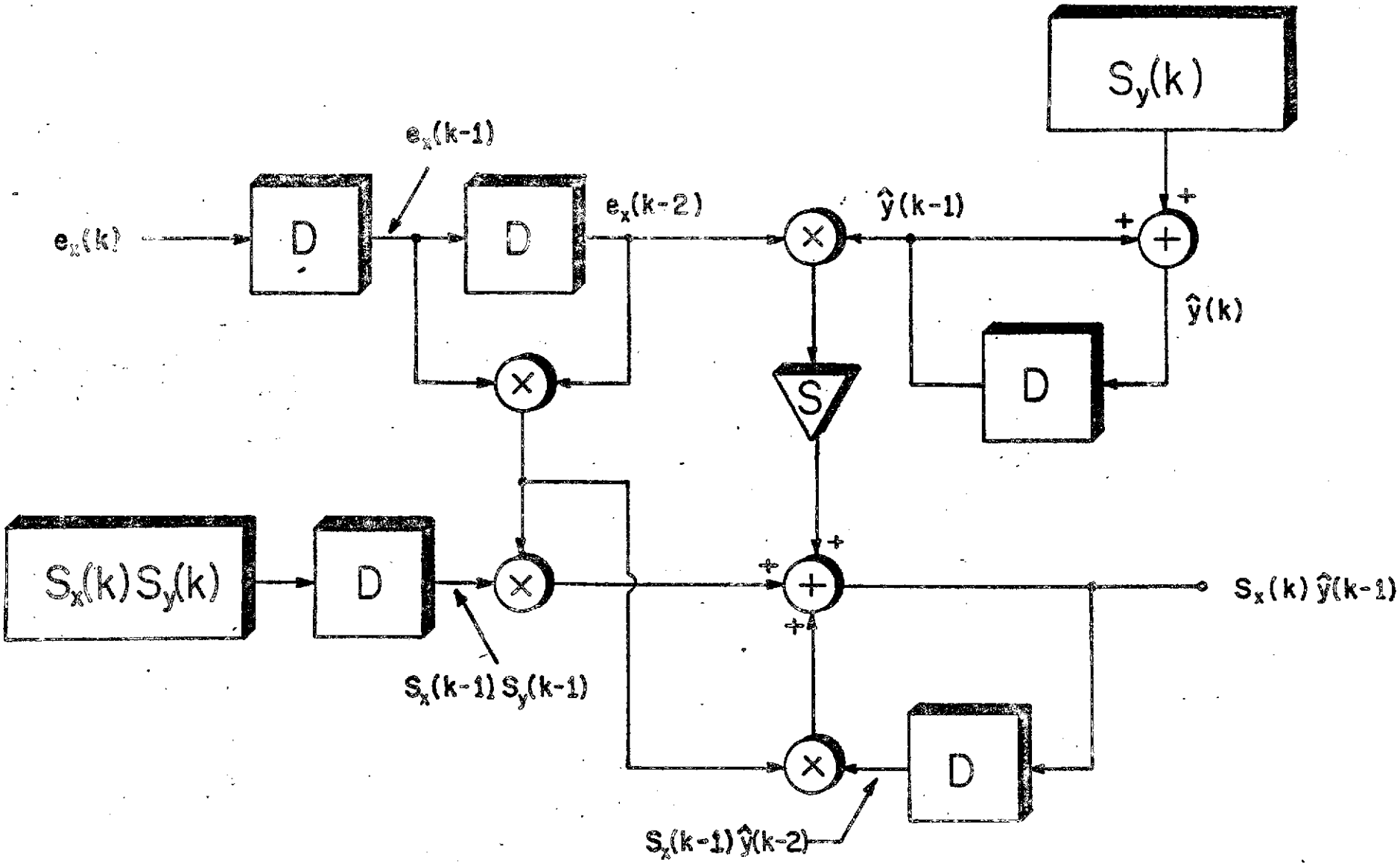


Figure 4c. Realization of $S_x(k)\hat{y}(k-1)$

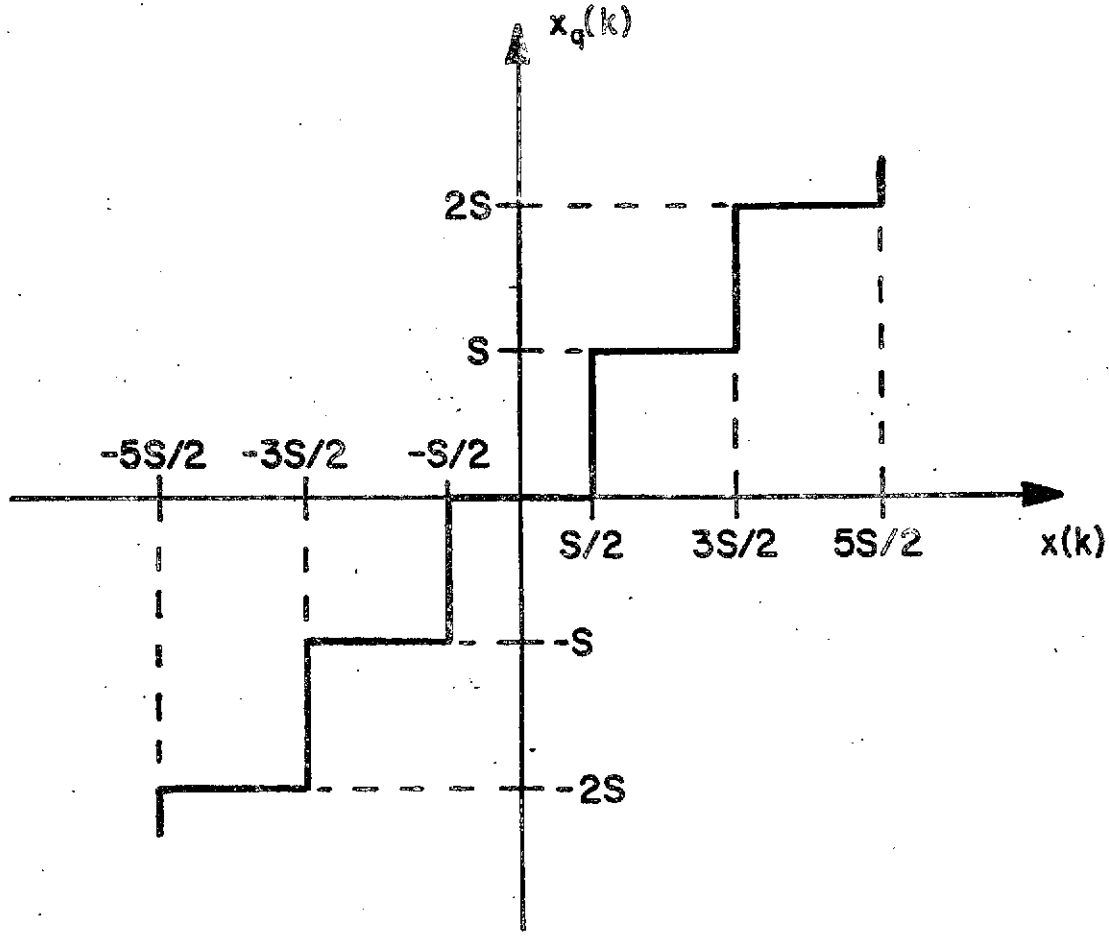


Figure 5. Input - Output Quantization Relationship

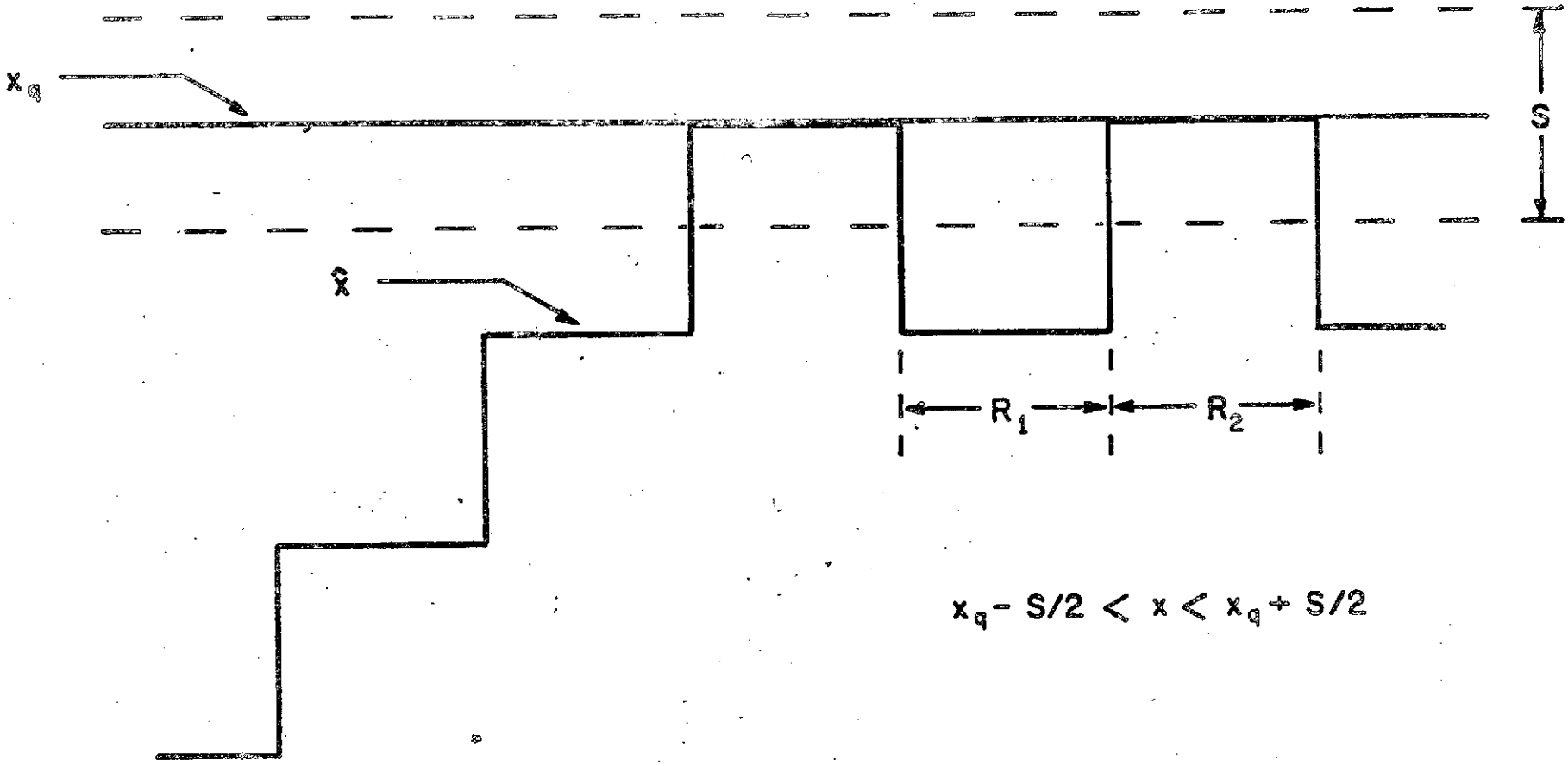


Figure 6. Response of a Linear DM to a Constant Input

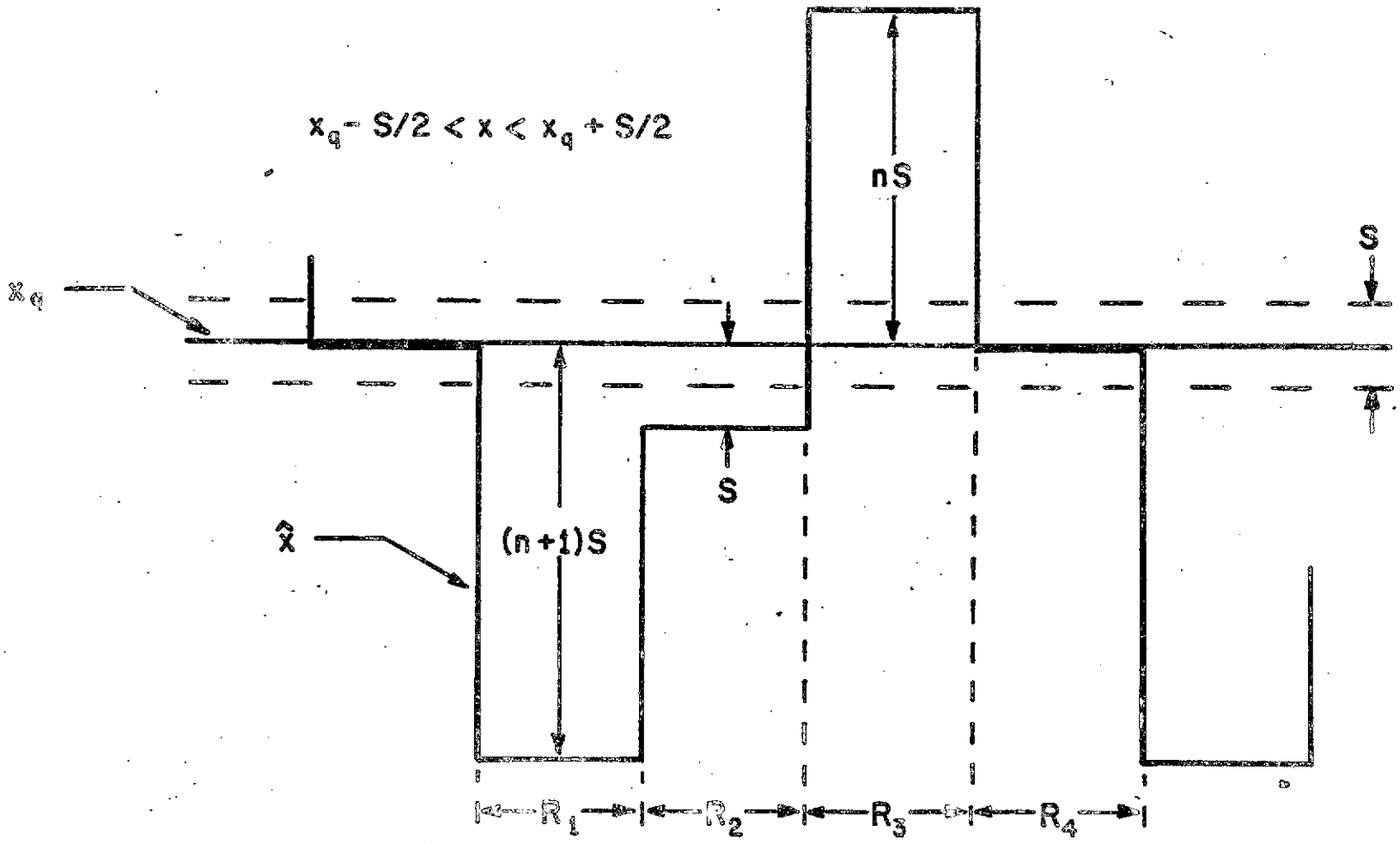


Figure 7. General Steady State Error Pattern of Enhanced Abate Mode Adaptive DM for a Constant Input

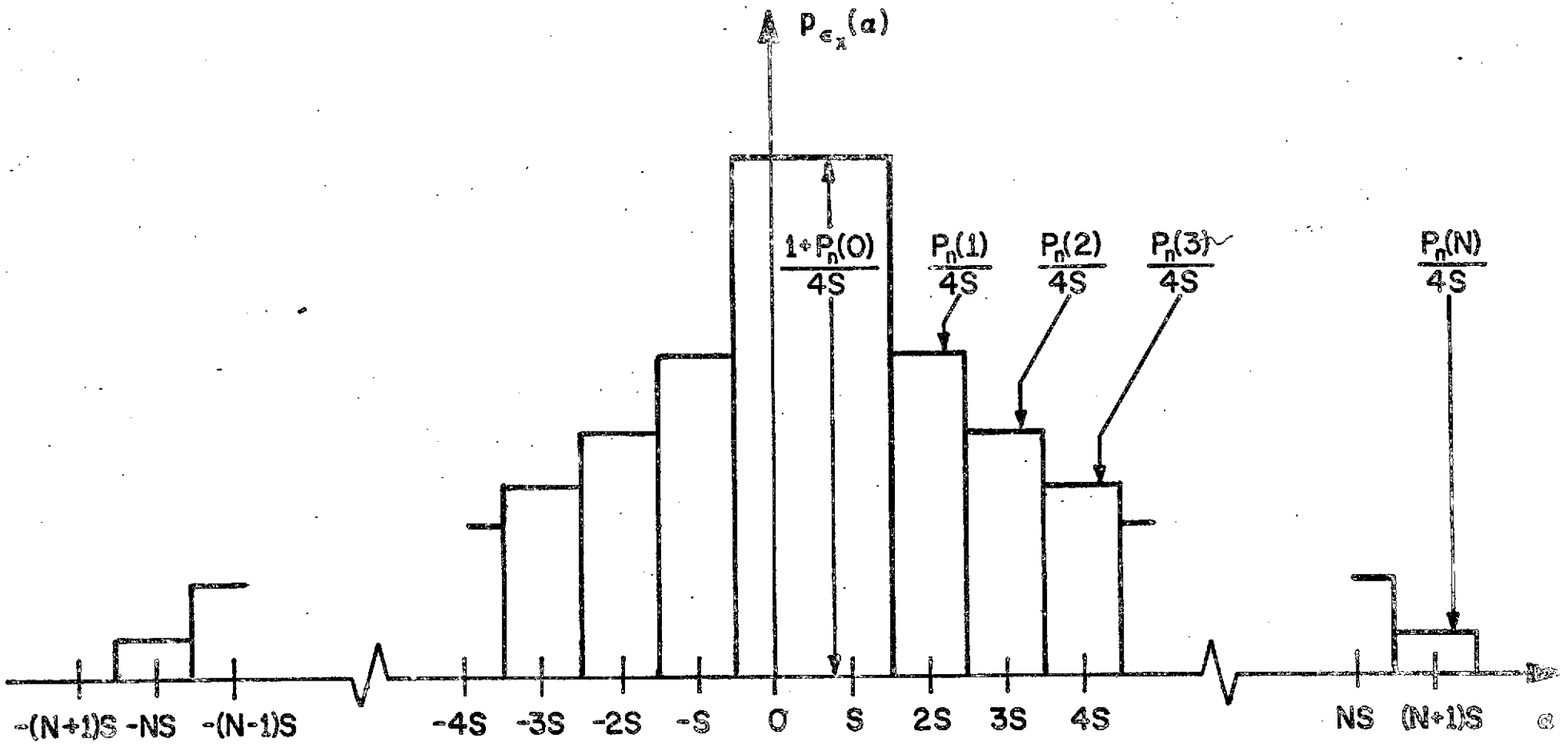


Figure 8. Probability Density Function of ϵ_x for Enhanced Abate Mode Adaptive DM with a Constant Input

II. New Area of Research

A Digital Phase Locked Loop With Nonlinear Processor

The Phase Locked Loop (PLL) is a nonlinear control system which has been under study by many investigators in the past few years as evidenced by the literature that has been published on the subject. Its applications range from bit synchronization to carrier tracking to FM demodulation.

Until recently most of the emphasis has been on analog phase locked loops as shown in Figure 1. This device consists of a phase detector, (PD), a filter, and a voltage controlled oscillator (VCO). The VCO is sometimes called a slave oscillator since, in steady state, its average frequency is that of the incoming signal ω_o.

The principle of operation is as follows: the phase of the VCO is an estimate of the phase of the incoming signal. This incoming signal plus noise and the VCO output are inputs to the PD. The output of the PD, $\psi(t)$ is a function of the phase error, $\epsilon(t) = \varphi_m(t) - \theta_v(t)$. This function $\psi(t) = f(\epsilon(t))$, is then filtered to form $V_D(t)$ which is the demodulated output voltage. Since $V_D(t)$ is proportional to the VCO frequency, $\theta_v(t)$, we can write

$$\theta_v(t) = G_{VCO} \int_{-\infty}^{t} f(\varphi_m(\lambda) - \theta_v(\lambda)) \cdot h(t-\lambda)\, d\lambda \qquad (1)$$

where $h(t)$ is the inverse Laplace transform of $H(s)$ the filter transform, $\varphi_m(t)$ is the modulated phase, $\theta_v(t)$ is the VCO phase estimate, and $f(x)$ is the PD function. Thus it is seen that in the PLL, the error signal is filtered to provide the necessary correction before it is fed back through the VCO to the PD.

Now, however, with an emphasis toward digital processing, investigation into various types of Digital Phase Locked Loops (DPLL) has been initiated. In a typical DPLL configuration as shown in Figure 2 we see that the incoming signal is first sampled and held and then A/D converted. This digital input signal is now digitally phase detected and filtered. The digital filter seems to smooth the error signal $\psi(k) = f(\varphi_m(k) - \theta_v(k))$.

In most PLL, the nonlinearity arises in the PD and the loop processor is a linear filter. However, there may exist advantages in using a nonlinear processor in a loop.

In particular, we are currently investigating the use of various algorithms used in digital Delta Modulation (DM) schemes. Consider, for example the DPLL system shown in Figure 3. Here the error signal $\psi(k)$ is processed by the nonlinear processor such that

$$e(k) = \text{sgn}\,[\psi(k)] \tag{2a}$$

and

$$V_D(k) = \Delta_o \sum_{i=0}^{k-1} e(i) = \Delta_o \sum_{i=0}^{k-1} \text{sgn}\,[\psi(i)], \tag{2b}$$

where Δ_o is the step size of the DM. Thus the DPLL response equation is given by the nonlinear recursive difference equation

$$\dot{\theta}_v(k+1) - \dot{\theta}_v(k) = G_{VCO}\,\text{sgn}\,[\psi(k)], \tag{3a}$$

where $\dot{\theta}_v(m)$ is the estimate of the incoming modulating signal at the m^{th} sampling instant and G_{VCO} is a constant gain. By letting the derivative at the k^{th} instant be equal to a difference i.e.,

$$\dot{x}(k) = [x(k) - x(k-1)] / \Delta T, \tag{3b}$$

we obtain the following:

$$\epsilon(k+1) - \epsilon(k) + \epsilon(k-1) = G_{VCO}\,\text{sgn}\,[\psi(k)] + [\varphi_m(k+1) - 2\varphi_m(k) + \varphi_m(k-1)]. \tag{3c}$$

It is seen that this DPLL system can be considered as a second order PLL using a digital filter employing 1-bit interval arithmetic. As such it is characterized by a nonlinear difference equation. This system is obtained when a processor used in a linear delta modulator is inserted in the loop. The nonlinearity is the sgn function which results in a binary phase detector.

As another means of nonlinear processing, the Song and Abate mode algorithms for Adaptive Delta Modulators are chosen.

The generalized equations for these modes are as follows:

A) Song Mode

$$V_D(k+1) = V_D(k) + \Delta V_D(k+1) \tag{4a}$$

$$\Delta V_D(k) = \alpha \mid \Delta V_D(k-1) \mid e(k-1) + \beta \mid \Delta V_D(k-1) \mid e(k-2) \tag{4b}$$

$$e(k) = \text{sgn} [\, \psi(k) \,] . \tag{4c}$$

Here $\Delta V_D(k)$ is the adaptive DM step size and α, and β are constants. Appropriate scaling can be introduced at the e(k)'s. Experimentally and theoretically it has been determined that $\alpha = 1$, and $\beta = \frac{1}{2}$, are optimum values as far as performance and hardware implementation (cost).

The DPLL response equation of this system is as follows:

$$\theta_v(k+1) - 2\theta_v(k) + \theta_v(k-1) = g_d \left\{ \mid \theta_v(k) - 2\theta_v(k-1) + \theta_v(k-2) \mid \cdot [\, e(k-1) + \tfrac{1}{2} e(k-2) \,] \right\} , \tag{4d}$$

where g_d is a scaling factor.

This type of DM was found to work well with video signals and therefore application of this loop to video transmission bears investigation.

In the Abate mode the following equations govern the system:

$$V_D(k) = V_D(k-1) + \Delta V_D(k) \tag{5a}$$

$$\Delta V_D(k) = \mid \Delta V_D(k-1) \mid e(k-1) + \Delta V_o\, e(k-2) \tag{5b}$$

$$e(k) = \text{sgn} [\, \psi(k) \,] . \tag{5c}$$

In the above ΔV_o is the minimum value of step size. The equation of response for the DPLL in this case is:

$$\theta_v(k+1) - 2\theta_v(k) + \theta_v(k-1) = \left\{ \left| \theta_v(k) - 2\theta_v(k-1) + \theta_v(k-2) \right| \cdot e(k-1) + G_{VCO} \Delta V_o e(k-2) \right\} \cdot \quad (5d)$$

This algorithm has been successful for voice and so this loop bears investigation for audio application.

There is a similarity to these equations. Techniques to solve the above nonlinear difference equations are now under investigation. Note also that these equations are for signals with no noise present. The response of these PLL in the presence of noise will also be investigated and performance on a SNR basis will be evaluated and compared to other present day systems.

Pull-in time and lock-on range will also be determined and an attempt made to minimize the pull-in time.

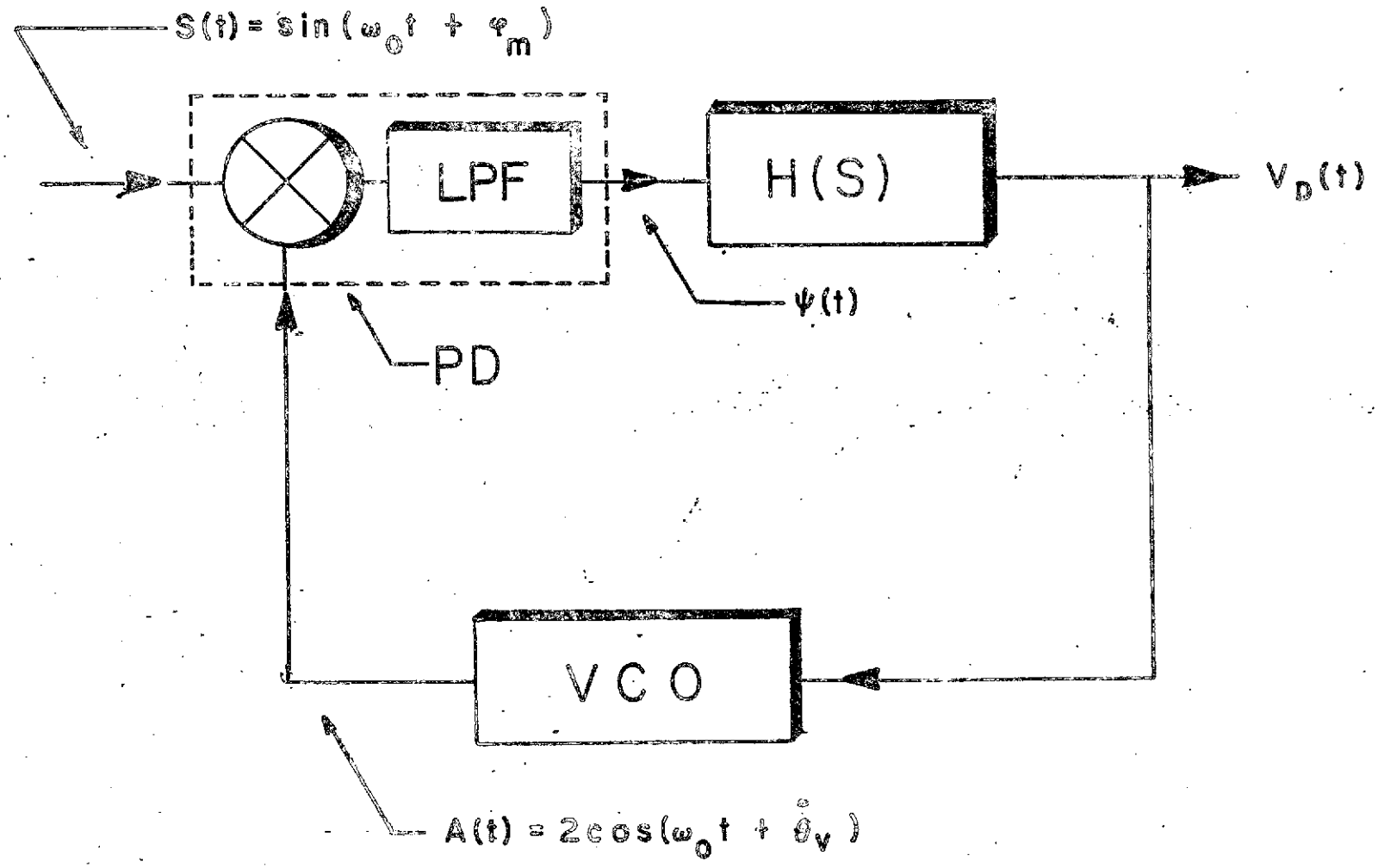


Figure 1 An Analog Phase Locked Loop

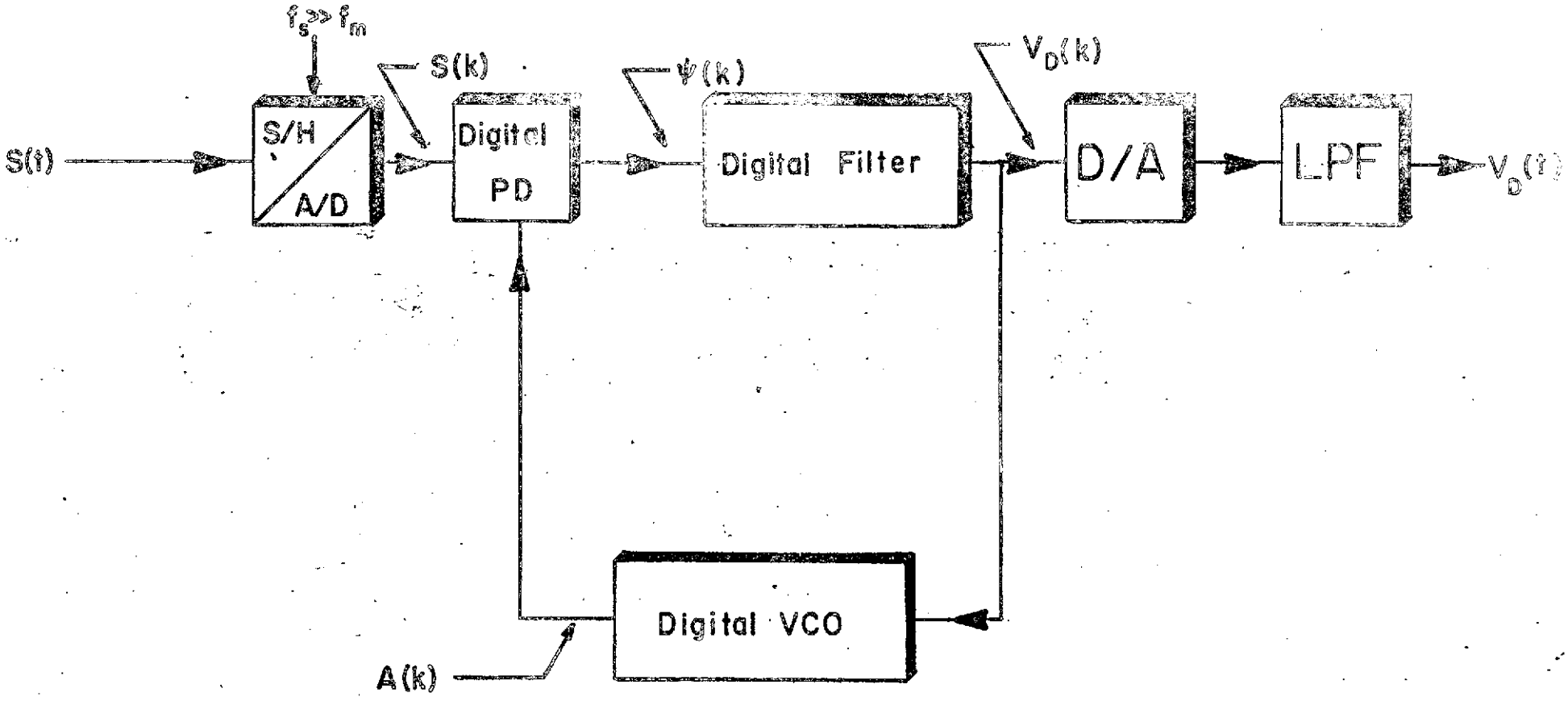


Figure 2 A Digital Phase Locked Loop

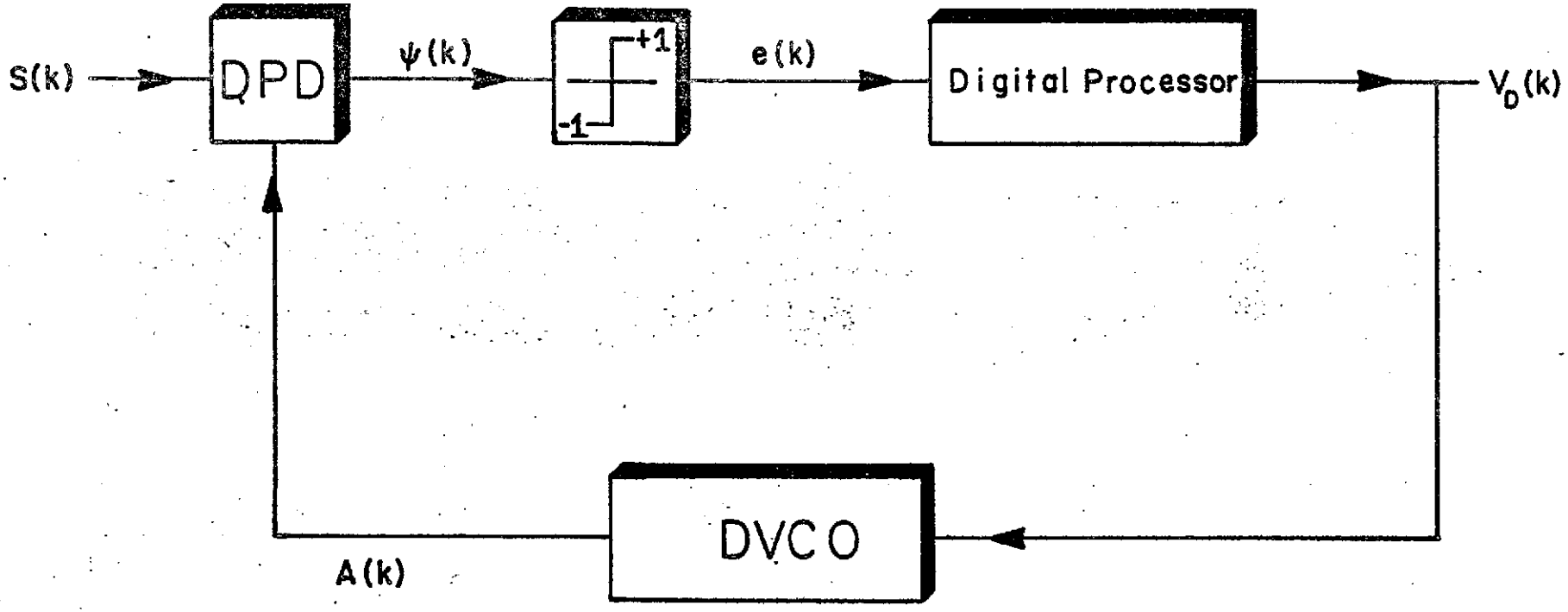


Figure 3 A DPLL with Nonlinear Error Processor

III. Doctoral Dissertation

"An All Digital Phase-Locked Loop for FM Demodulation"

by John F. Greco

IV. Papers Published and Presented

1. "An All Digital Phase-Locked Loop for FM Demodulation," J.F. Greco and D. L. Schilling, pp. 43.37-43.41, ICC-73, June 1973.
2. "Overshoot Suppression in Adaptive Delta Modulator Links for Video Transmission," L. Weiss, I. Paz and D. L. Schilling, pp. 6D.1-6D.6, NTC-73, Nov. 1973.
3. "Delta Modulation: With Applications to Video," D. L. Schilling, Seventh Hawaii Int'l Conference on System Sciences, Jan. 1974.
4. "Technique for Correcting Transmission Errors in Delta Modulation Channels with Application to Video Signal Processing," Z. Ali, I. Paz, D. L. Schilling and N. Scheinberg, ICC-74, June 1974.
5. "Response of an All Digital Phase-Locked Loop," J. Greco, J. Garodnick and D. L. Schilling, IEEE Trans. on Communications, June 1974.

The papers which were available are reproduced below.

AN ALL DIGITAL PHASE-LOCKED LOOP FOR FM DEMODULATION

John Greco and Donald L. Schilling
Department of Electrical Engineering
City College of The City University of New York
New York, New York 10031

Abstract

An all digital phase-locked loop for FM demodulation is presented. The unit operates as a real time special purpose digital computer and employs a square wave voltage controlled oscillator. Design procedures are given for a first, second, and third order loop; the design reflects the influence of the square wave oscillator as well as quantization limitations. In an attempt to obtain information about the threshold of the digital phase-locked loop, the response to artificial input noise spikes is examined.

1. Introduction

The block diagram of the digital phase-locked loop (DPLL) is shown in Fig. 1. All signals to the right of the A/D converter are binary words and all computations within the DPLL are digital. The digital filter is either a proportional path, a proportional plus integral path, or a proportional plus integral plus double integral path yielding the first, second, or third DPLL respectively. To simplify the implementation, the filter gains are restricted to be $1/2^N$, with N an integer, so that coefficient multiplication is reduced to shifting the binary word. To avoid a binary multiplication of the input x_k and VCO output w_k at the phase detector, the digital VCO waveform is a square wave having values ± 1. The multiplication is then reduced to a simple logic operation. An algorithm is developed (1) to determine the correct output of the digital VCO and an implicit VCO gain is introduced (1) : $G_{VCO} = \pi 2^N/5$, N = integer.

2. Digital Phase-Locked Loop Equation

To avoid distortion in the sampled incoming FM signal, the sampling frequency, f_s, is chosen according to

$$f_s = 2m\,B \tag{1a}$$

$$nf_s = f_c - B/2 \tag{1b}$$

where B = IF bandwidth

$m \geq 1$

n = integer

f_0 = carrier frequency.

The VCO output is given by

$$\omega_k = Sq\,(k\frac{\pi}{2m} + \hat{\varphi}k) \tag{2}$$

where $\hat{\varphi}_k$ = VCO phase,

$$Sq\,(x) = \begin{cases} +1, & 0 \leq x < \pi \\ -1, & \pi \leq x < 2\pi \end{cases}$$

and Sq (x) is periodic with period 2π. The DPLL output y_k is the derivative of the VCO phase $\hat{\varphi}_k$; in our digital scheme the derivative is computed as the difference:

$$y_k = \hat{\varphi}_{k+1} - \hat{\varphi}_k \tag{3}$$

Finally, the output y_k is determined from the phase error signal e_k by the digital filter characteristic. For the first order DPLL, $y_k = g\,e_k$ and combining Eqs (2) and (3), we obtain the difference equation for the VCO phase:

$$\hat{\varphi}_{k+1} = \hat{\varphi}_k - 2G\cos\,(k\frac{\pi}{2m} + \varphi_k)\,Sq\,(k\frac{\pi}{2m} + \hat{\varphi}_k) \tag{4}$$

where G = loop gain. This nonlinear difference equation is extremely difficult to analyze and so a linearized model is introduced.

3. Linearized Model

The function Sq (·) contains odd harmonics and the product of input and square wave produces harmonics at 0, $f_s/2m$, $2f_s/2m$, . . . , $Nf_s/2m$, . . . Hz. Of course, the zero harmonic is the only useful term as it contains the phase error information. Hence, we would like to have the other harmonics fall outside the DPLL bandwidth. But this is impossible as the $(2m)^{th}$ harmonic is at f_s Hz which is equivalent to 0 Hz. That is, the aliasing produced by sampling introduces a term at 0 Hz. This term has amplitude on the order of $1/4m$, and so it appears that we should choose m large. However, choosing m large results in the harmonic at $f_s/2m$ being introduced into the DPLL bandwidth. The constraint we impose is to have this harmonic fall outside the DPLL bandwidth. If the input frequency deviation is Δf, the bandwidth of this harmonic is approximately $4\,\Delta f$, and so the constraint is

$$f_s/\,2\,m - 2\,\Delta f \geq B_L \tag{5}$$

with B_L = DPLL bandwidth.

If we can choose $m \geq 3$, the $(2m)^{th}$ harmonic will be at least 20 dB below the fundamental and its contribution to the output and VCO phase can be neglected.

With Eq (5) satisfied we introduce the linearized model of the DPLL by using the VCO fundamental to generate the phase error. This fundamental has amplitude $4/\pi$, and assuming $|e_k| << \pi/2$, we arrive at the linearized model shown in Fig. 2, where G_{VCO} = implicit VCO gain.

4. Phase Error Contraint

The phase error e_k should be small so that we operate on the linear portion of the phase characteristic; i.e., $\sin e_k \simeq e_k$. The error in this approximation is referred to as harmonic distortion. However, e_k is represented by a finite number of bits and a small e_k will be lost in the quantization noise. Hence, an optimum phase error exits for which signal-to-harmonic distortion and signal-to-quantization noise are equal. These quantities are computed (assuming e_k uniformly distributed over its range) and plotted against $(e_k)_{max}$ for 10-bit quantization and 12-bit quantization in Fig. 3. The intersections give the optimum value of $(e_k)_{max}$:

Number of Bits	$(e_k)_{max}$ (volts)
10	0.35
12	0.23

5. Design of the First Order DPLL

An input frequency offset of Δf produces a phase error

$$e_k = \frac{2\pi\,\Delta f/f_s}{G} \tag{6}$$

The DPLL is constructed using 10-bit arithmetic; there-

fore

$$\frac{2\pi \Delta f/f_s}{G} < 0.35 \qquad (7)$$

The DPLL bandwidth B_L is given approximately by

$$B_L \simeq f_s \frac{2}{\pi^2} G \qquad (8)$$

where the approximation is valid for $\frac{4}{\pi}G << 1$. Utilizing Eq. (8) in Eq. (5),

$$\frac{f_s}{2m} - 2\Delta f \geq f_s \frac{2}{\pi^2} G \qquad (9)$$

Combining with Eq. (7) yields the design equation

$$m \leq \frac{f_s}{11.28\ \Delta f} \qquad (10)$$

The DTL and TTL logic employed necessitated the value $f_s = 50$ kHz to provide a sufficient computation interval. Choosing $\Delta f = 600$ Hz we obtain $m \leq 7.4$. We shall use $m = 8$ (the VCO algorithm is simplified when m is a power of 2). We further obtain from Eq (7) $G \geq .214$ and $B_L = 2.14$ kHz. The choice $m = 8$ invalidates Eq (5); we shall choose

$$G = 0.157 = \pi/5 \cdot 2^3 \qquad (11)$$

and allow $(e_k)_{max} = 0.46$. Then $B_L = 1.57$ kHz (the approximation of Eq. (8) is conservative; the actual bandwidth is 1.8 kHz) and Eq. (5) is satisfied.

Summarizing, the parameters for the first order DPLL are:

$f_s = 50$ kHz
$\Delta f = 600$ Hz
$G = 0.157 = \pi/5 \cdot 2^3$

6. Design the Second Order DPLL

The digital filter is augmented to a proportional plus integral filter having transfer function H(z) given by

$$H(z) = g_1 + g_2 \frac{1}{1 - z^{-1}} \qquad (12)$$

where g_1 = proportional path gain
g_2 = integral path gain.

We shall use the same sampling frequency and frequency deviation computed for the first order DPLL.

The second DPLL tracks a frequency offset with zero phase error. Therefore we consider sinusoidal modulation: $\hat{\varphi}_k = \beta \sin k\, 2\pi f_m/f_s$. If $f_m << f_s$, the phase error amplitude is

$$(e_k)_{max} \simeq \frac{2\pi\ \Delta f/f_s}{G_2}\ 2\pi\ f_m/f_s \qquad (13)$$

where G_2 = gain of the integral loop. Choosing $f_m = 200$ Hz ($\beta = 3$) we obtain from the phase error restriction $G_2 \geq .0055$. We choose initially $G_2 = .009818 = \pi/5 \cdot 2^6$.

Now we have m and G_1 at out disposal to meet Eq.(5). But we wish to achieve more than simply satisfying this equation--we seek a second order DPLL which provides threshold improvement over the first order DPLL. To gain insight into the loop's behavior near threshold, we introduce an artifical spike to the DPLL and observe the response (VCO phase). Obviously, if the VCO phase follows the input spike, the spike is reproduced at the output.

The model used for the spike is a sinusoidal increase of 2π radians in the input phase:

$$(\varphi_k)_{spike} = 2\pi(\tfrac{1}{2} - \tfrac{1}{2}\cos k\pi / K) \qquad (14)$$

The spike duration, K/f_s, is chosen as the reciprocal of the IF bandwidth; this is the fastest spike possible. This spike is superimposed on constant or sinusoidal modulation, producing a positive spike when the input frequency is at the left extreme of the IF bandwidth. The nonlinear difference equation is solved on a computer for the VCO phase and the solution is examined to determine whether or not the loop follows the spike. Note that the linearized model is not valid when a spike appears as the phase error becomes large. Also note that the carrier amplitude remains constant during the spike; this represents a worst case as in actuality the amplitude decreases.

The results are displayed in Fig. 4, where the spike responses are displayed as a function of loop gains. Fig. 4 (a) is the case of the spike superimposed upon an input frequency deviation of 600 Hz; Fig. 4 (b) is for sinusoidal modulation with $f_m = 200$ Hz and $\beta = 3$. They clearly show that $m = 8$ is unsatisfactory as the loop becomes unstable for almost all values of G_1. The case $m = 4$ is not much better. If we decrease the integral gain to $\pi/5 \cdot 2^7$ (and consequently increase the phase error) then we can choose $G_1 = \pi/5 \cdot 2^4$ or $\pi/5 \cdot 2^5$ or $\pi/5 \cdot 2^6$. The final choice depends on satisfying Eq. (5) and also on the linearized input-output loop transfer function. As we are interested in FM demodulation, the DPLL transfer function should be as flat as possible out to f_m. Examing these conditions leads to the choice $G_1 = \pi/5 \cdot 2^4$.

Summarizing, the second order DPLL gains are

$$G_1 = \pi/5 \cdot 2^4 \qquad G_2 = \pi/5 \cdot 2^7$$

7. Design of the Third Order DPLL

A double integral filter is added to the previous filter of Eq. (12):

$$H(z) = g_1 + g_2 \frac{1}{1 - z^{-1}} + g_3 \left(\frac{1}{1 - z^{-1}}\right)^2 \qquad (15)$$

yielding the third order DPLL. For sinusoidal modulation, the phase error amplitude is approximately given by

$$(e_k)_{max} \simeq \frac{(2\pi f_m/f_s)^2}{G_3}\ 2\pi\ \Delta f/f_s \qquad (16)$$

where G_3 = gain of the double integral path. Imposing the phase error restriction yields (using $\Delta f = 600$ Hz and $f_m = 200$ Hz) $G_3 \geq 0.000132$. We choose $G_3 = 0.000154 = \pi/5 \cdot 2^{12}$ and $m = 4$. The values of G_1 and G_2 are chosen from th spike response and linearized loop characteristic. The spike responses illustrated in Fig. (5) show that there exist several pairs of gains G_1, G_2 for which the third order DPLL suppresses the input spike, both for constant and sinusoidal modulation. To narrow the choice, a spike lasting $2/(\text{IF bandwidth})$ seconds is introduced and the final choice is

$$G_1 = \pi/5 \cdot 2^5 \qquad G_2 = \pi/5 \cdot 2^3 \qquad G_3 = \pi/5 \cdot 2^{12} \qquad (17)$$

These gains produce a linearized characteristic which is flat within 0.5 dB to 200 Hz and has a 730 Hz bandwidth, thereby satisfying Eq. (5).

8. Conclusions

An all digital phase-locked loop has been presented and a design procedure for a first, second, and third order loop is developed. The procedure involves quantization effects and response to artificial input noise spikes. Experimental results based on this design will be presented.

References

1. An All Digital Phase-Locked Loop for FM Demodulation, J. Greco, J. Garodnick, D. L. Schilling International Conference on Communications, Phil. Pa., June 1972.

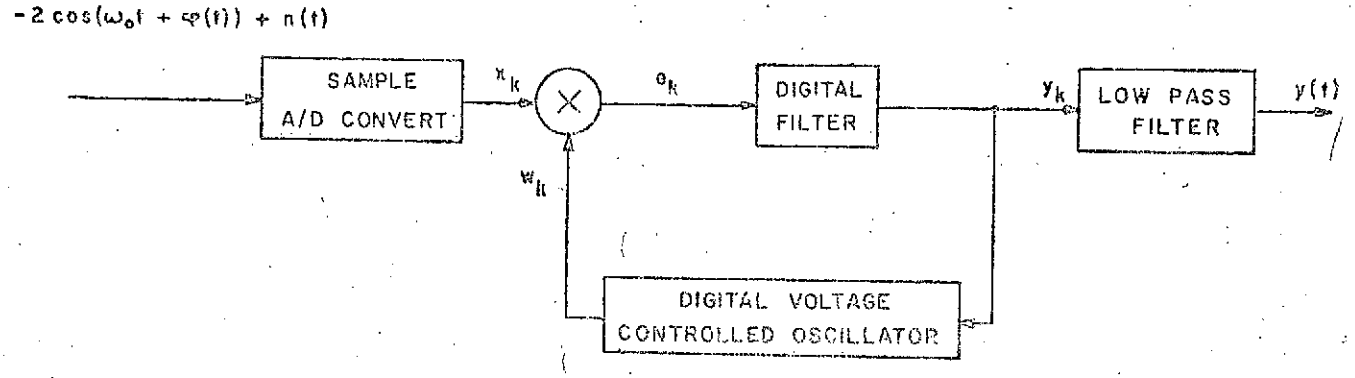


Fig. 1 Block diagram of the all digital phase-locked loop

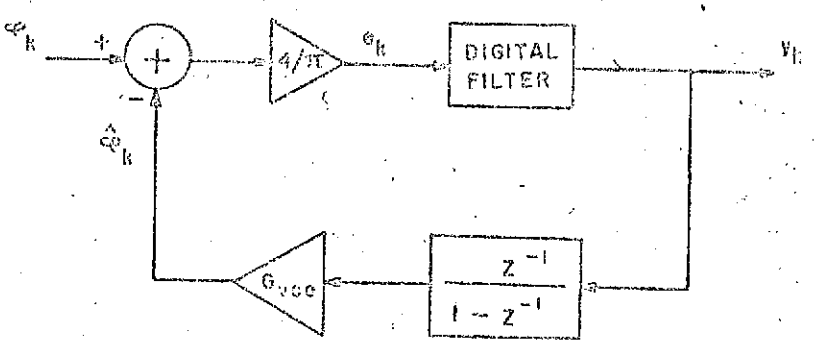


Fig. 2 Block diagram of the linearized model of the digital phase-locked loop

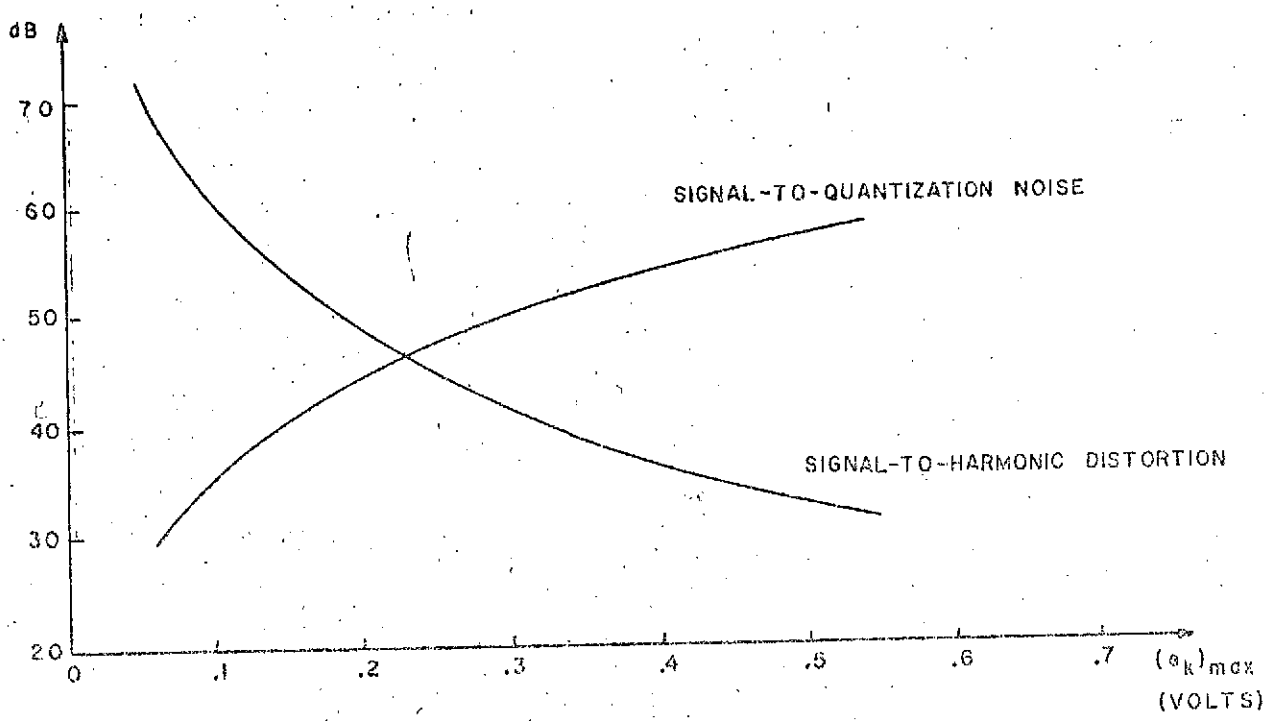


Fig. 3a Signal-to-quantization noise and signal-to-harmonic distortion as a function of maximum phase error for a 12-bit A/D converter.

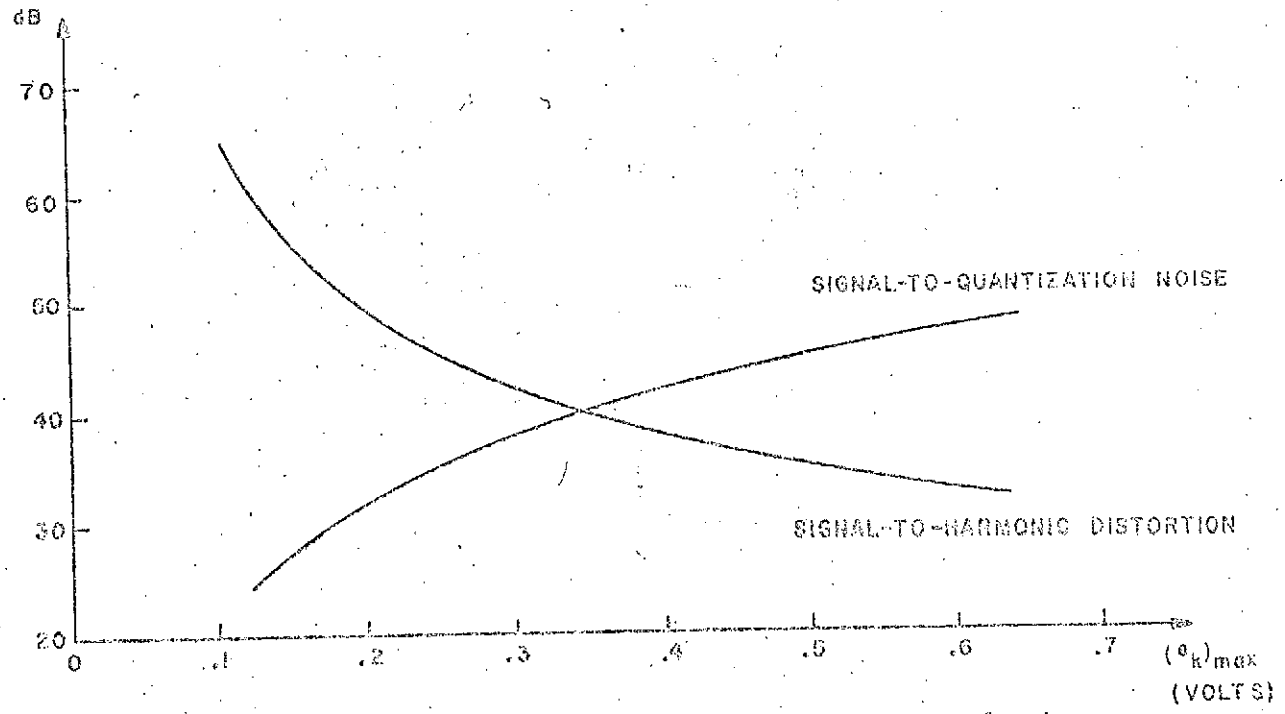
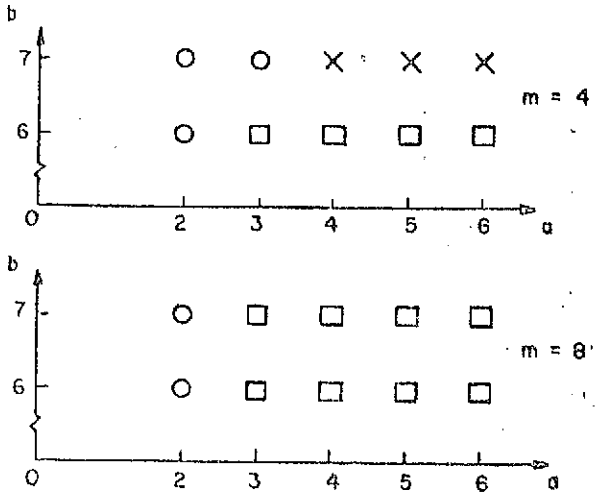


Fig. 3b Signal-to-quantization noise and signal-to-harmonic distortion as a function of maximum phase error for a 10-bit A/D converter.



$G_1 = \pi/5 \cdot 2^a$ $\quad G_2 = \pi/5 \cdot 2^b$

× LOOP DOES NOT FOLLOW INPUT SPIKE

O LOOP FOLLOWS INPUT SPIKE

□ LOOP BECOMES UNSTABLE

Fig. 4a. Second order DPLL spike response; the spike is superimposed on a 600 Hz frequency offset.

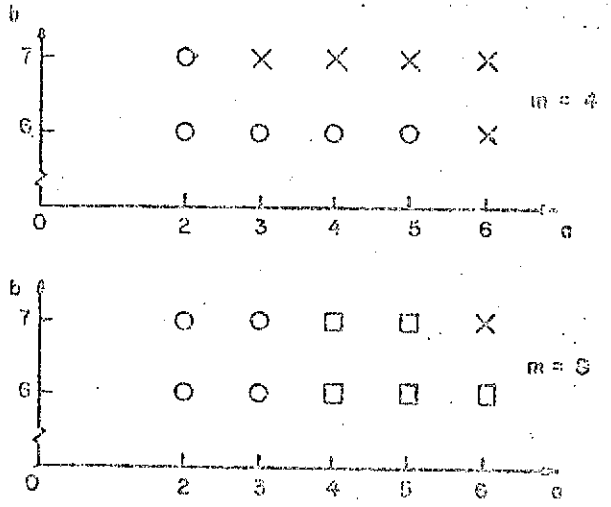
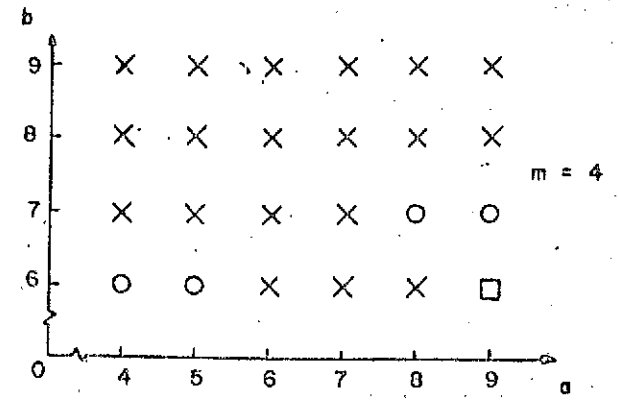
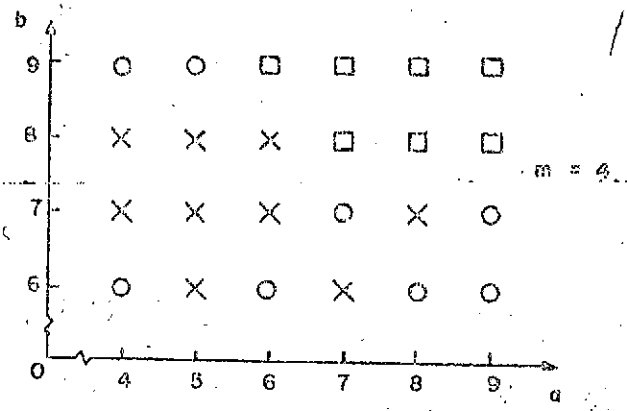


Fig. 4b. Second order DPLL spike response; the spike is superimposed on sinusoidal modulation with $f_m = 200$ Hz and $\beta = 3$.



CONSTANT MODULATION $\Delta f = 600$ Hz



SINUSOIDAL MODULATION $\Delta f = 600$ Hz $\beta = 3$

$G_1 = \pi/5 \cdot 2^a$ $\quad G_2 = \pi/5 \cdot 2^b$ $\quad G_3 = \pi/5 \cdot 2^{12}$

Fig. 5. Third order DPLL spike response.

OVERSHOOT SUPPRESSION IN ADAPTIVE DELTA MODULATOR LINKS FOR VIDEO TRANSMISSION

by

L. Weiss, I. Paz, D. L. Schilling
Department of Electrical Engineering
The City College of C.U.N.Y.

ABSTRACT

An overshoot suppression scheme to improve the performance of the Digital Song Adaptive Delta Modulator for picture transmission is described. The overshoot suppression algorithm has been verified using computer simulation on a PDP-8. It is also shown that the additional hardware required for the actual implementation of the algorithm is simpler than those encountered in the literature, and gives better signal tracking accuracy.

I. INTRODUCTION

Delta Modulation is the name of the encoding process in a Digital Communications Link that allows for only changes in the input signal to be processed and transmitted. Using this procedure the link is usually required to transmit only one bit per sample, resulting in an improved bandwidth utilization efficiency over that obtained with PCM systems.

Briefly, a delta modulator operates as follows. The amplitude of the signal to be transmitted is sampled periodically and compared to an estimated value. The estimated signal is obtained by incrementing the previous estimate at each sampling time by a discrete amount called a step size. The sign of the difference between the signal and its estimate is used to decide if the previous estimate should be increased or decreased. This sign information (one bit per sample) is also transmitted over the channel to the decoder. The decoder uses this bit stream to construct the signal estimate.

A video signal is characterized by discontinuities of large amplitude and very short rise time. This corresponds to abrupt changes in shades in the picture content. A linear delta modulator is limited in its ability to track sudden input changes by its fixed step size. The magnitude of the steps is bounded by the permissible granular noise in constant shade regions, see Fig 1a. Shade contrast is thus degraded by the so called "slope-overload-noise" introduced by the delta modulators inability to rapidly follow the signal discontinuity. To alleviate this condition, while maintaining the permissible granular noise level, it is desirable to make the step sizes small initially but allow them to increase quickly in some nonlinear fashion when tracking a rapidly varying input, see Fig 1b. This is done in an adaptive delta modulator [1,2].

The sharp rises in a video signal are usually followed by regions of constant level due to regions of uniform shade in the picture. Thus while alleviating slope overload problems, an adaptive delta modular introduces the possibility of large overshoots when the tracked level is finally reached. Furthermore, the overshoot is followed by a transient oscillatory response until the delta modulator finally locks onto the tracked signal level. These effects are shown in Fig 1(b).

Both the overshoot and the subsequent recovery time are undesirable attributes of an adaptive delta modulator. Reducing the step sizes decrease the possible overshoot amplitudes and shortens the recovery time. This, however, augments slope overload. A trade-off therefore exists between overshoot amplitude and recovery time versus slope-overload-noise in an adaptive delta modulator.

Overshoot suppression is a scheme to sharply limit the overshoot amplitude and reduce the subsequent recovery time. This is done without reducing the step sizes until overshoot is imminent. The trade-off between overshoot amplitude and recovery time versus slope-overload-noise is thus relaxed.

An overshoot suppression scheme has recently been suggested in the literature [3]. This scheme, however, uses a look-up table in which arbitrary values for the step sizes are suggested. It could therefore not serve the optimal delta modulator we are investigating in which the step sizes are obtained by explicit mathematical expressions. Furthermore, the maximum step size is limited in the above scheme by overshoot considerations. However, in the Song Delta Modulator, the step sizes can continually increase with the overshoot suppression scheme described below, thus yielding better signal to slope-overload-noise ratio. Moreover, the amount of equipment involved in implementing the proposed overshoot suppression algorithm is very modest in comparison to the equipment needed to implement other proposed schemes, and could fit into any adaptive delta modulator in which the next step size is explicity calculated. It is also flexible as to the amount of overshoot suppression it can perform and trade-offs between conflicting factors can be accurately set as may be necessary.

II. VIDEO TRANSMISSION CHARACTERISTICS OF THE DIGITAL-SONG-VARIABLE-STEP-SIZE DELTA MODULATOR

Figure 2 shows the structure of the digitally implemented optimum adaptive delta modulation system referred to in this paper. Briefly, its operation is as follows:

The input signal S(t) is sampled and A to D converted to give S_k. S_k is then compared to its estimate, X_k, generating a sign-bit e_k, with

$$e_k = \text{sgn.}\,(S_k - X_k) \tag{1}$$

where

$$x_k = X_{k-1} + \Delta_k \tag{2}$$

The step size at the k^{th} sampling instant is

$$\Delta_k = g_1(e_{k-1}, \Delta_{k-1}) + g_2(e_{k-2}, \Delta_{k-1}) \tag{3}$$

Thus the k^{th} step size depends on the previous step size, and the previous two sign bits. The g_1 and g_2 function characteristics are shown in Fig 3 indicating that:

The research presented in this paper was partially supported by NASA grants NGR 33-013-003 and NGR 33-013-077.

$$\Delta_k = \begin{cases} |\Delta_{k-1}|(\alpha e_{k-1} + \beta e_{k-2}), & |\Delta_{k-1}| \geq 2\Delta_o \\ 2\Delta_o e_{k-1}, & |\Delta_{k-1}| < 2\Delta_o \end{cases} \quad (4)$$

where Δ_o is the minimum possible step size. The special region, $|\Delta_{k-1}| < 2\Delta_o$, is needed to prevent a dead zone at the origin. The decoder is just the feedback portion of the encoder. It reconstructs the approximation X_k from the e_k sign bit stream. X_k is then D/A converted and low pass filtered to give $\hat{S}(t)$, the estimate of the transmitted signal.

In video processing, S(t) will contain many large discontinuities of very short rise time followed sometimes by constant levels. Thus edge response is extremely important in video. To permit $\hat{S}(t)$ to approximate rapid rises, i.e., minimize slope-overload-noise, rapidly increasing step sizes are required. This can be accomplished by increasing α, β, as well as Δ_o in (4). Gains in reducing slope-overload-noise are made at the expense of large overshoots and long subsequent recovery times. Furthermore, it can be seen in Fig 1(b) that good steady state response, i.e., small amplitude oscillations about a constant level in S(t), requires small Δ_o. It can also be shown that the delta modulator becomes unstable if α and β are made too large. Thus in choosing α, β, and Δ_o a trade-off must be made between slope overload noise versus overshoots, recovery time, and steady state response while maintaining delta modulator stability.

The addition of overshoot suppression to the adaptive delta modulator permits α and β to be increased while decreasing Δ_o. In this way, slope overload as well as small steady state response requirements can be met simultaneously. Impending instabilities due to large α and β are also inhibited and, obviously, overshoots and subsequent recovery times are minimized.

III. THE PROPOSED OVERSHOOT SUPPRESSION ALGORITHM

The Overshoot Suppression Algorithm may be understood by considering the four cases shown in Fig 4 in which an overshoot or an undershoot occurs. In Fig 4(a) an overshoot occurs at sampling time k-1 followed immediately by an undershoot at k. For this case it is easy to show that the delta modulator considered will approach its steady-state condition rapidly. This is not the case in Fig 4(b) where the overshoot is larger than in (a) and X_k is greater than S_k. Consequently an undershoot occurs at K+1 or later and with an amplitude larger than in (a). This occurs because the step sizes begin increasing again after the first reversed step. Thus it will take many more sampling periods to reach steady state in (b) than in (a). The algorithm is therefore implemented only when case (b) occurs. Note that Fig 4(c) and 4(d) depict undershoots corresponding to the overshoots in 4(a) and 4(b) respectively. Action to prevent excessive undershoots is thus taken only for case (d).

The occurrence of cases (b) and (d) can be recognized by examining the sequence $(e_{k-3}, e_{k-2}, e_{k-1}, e_k)$. The fingerprint of (b) is (+1, +1, -1, -1), while that of (d) is (-1, -1, +1, +1). When either sequence is encountered action is taken to prevent overshoot or undershoot. The corrective action entails decreasing the stored values of Δ_{k-1} and hence X_{k-1} and X_k. Case (b) is thus transformed into a case (a) situation and the same for (d) and (c) respectively. The shape of the modified waveform actually depends on the amount by which Δ_{k-1} is decreased. The simplest scheme is one where Δ_{k-1} is replaced by half of its original value. We may allow for more rapidly increasing step sizes, Δ_k, i.e., larger α and β (see Eq 4) , as long as Δ_{k-1} is replaced by a smaller fraction of its original value when overshoot suppression is employed. Thus there is a faster initial rise coupled with a very sharp braking action just before the desired level is reached. However, since the braking action occurs close to the desired level there is nominal slope overload degradation. Indeed, there is an overall decrease in slope overload noise due to the more rapid increase in the initial step sizes.

Now the Oveshoot Suppression Algorithm is applied to the adaptive delta modulator operating in the Song Mode, i.e., $\alpha=1$, $\beta=0.5$. It is shown elsewhere [4] that usable video transmission can be obtained using these parameters even without overshoot suppression. With the addition of the suppression algorithm video reproduction should be much improved.

The salient features of the Song Mode response are now summarized. In approaching a level from above or below as in Fig 4, each step size is 1.5 times the previous one (see Eq 4 for $\alpha=1$, $\beta=0.5$). When a direction reversal occurs, as at sampling time k in Fig 4, then the first step size following the reversal is one half the previous step size, i.e., $\Delta_k = -\frac{1}{2}\Delta_{k-1}$ (see Eq 4). Thus, in Fig 4(b) we have

$$X_k \geq X_{k-1} - \tfrac{1}{2}\Delta_{k-1} = X_{k-2} + \tfrac{1}{2}\Delta_{k-1} \quad (5)$$

The inequality sign is needed due to the fixed point arithmetic employed in the digital implementation. Also in Fig 4(b) $X_{k-2} < S_k < X_k$. To implement overshoot suppression set $(\Delta_{k-1})' = \frac{1}{2}\Delta_{k-1}$, where the prime refers to the new values after the overshoot suppression algorithm has been implemented. Therefore,

$$(X_{k-1})' = X_{k-2} + (\Delta_{k-1})' = X_{k-2} + \tfrac{1}{2}\Delta_{k-1} \quad (6)$$

Next, set

$$(\Delta_k)' = \Delta_k = -\tfrac{1}{2}\Delta_{k-1} \quad (7)$$

Thus

$$(X_k)' = X_{k-1})' + (\Delta_k)' = X_{k-2} \quad (8)$$

Hence, Fig 4(b) has been transformed into Fig 5, with undershoot occurring at k rather than k+1 or later. It should be evident from Fig 5 even without a detailed explanation of the worst case that the overshoot has been at best entirely eliminated or at worst cut in half depending on whether S(t) is closer to $(X_{k-1})'$ or X_{k-2} respectively. Figure 5 also shows that the recovery time is greatly reduced, since the delta modulator locks onto S(t) very rapidly after sampling time k. Note also that now $(e_k)' = \text{sgn.}\,(S_k - (X_k)') = +1$, whereas in Fig 4(b) $e_k = \text{sgn.}\,(S_k - X_k) = -1$.

The above overshoot suppression scheme is now summarized in the form of an algorithm by considering a typical cycle of the now modified delta modulator.

Step 1: Generate S_k

Step 2: Calculate $\Delta_k = g_1(e_{k-1}, \Delta_{k-1}) + g_2(e_{k-2}, \Delta_{k-1})$

Step 3: Calculate $X_k = X_{k-1} + \Delta_k$

Step 4: Calculate $e_k = \text{sgn}.(S_k - X_k)$ and transmit this bit.

In the delta modulator without over shoot suppression this would complete the cycle. That is, k is next updated and steps through 4 are repeated. To implement overshoot suppression the following additional steps are needed.

Step 5: If $e_{k-3} = e_{k-2} = +1$, and $e_{k-1} = e_k = -1$, set V=1.
If $e_{k-3} = e_{k-2} = -1$, and $e_{k-1} = e_k = 1$, set W=1.

Step 6: If V≠1 and W≠1 go to 7, otherwise set

(a) $(\Delta_{k-1})' = \frac{1}{2}\Delta_{k-1}$

(b) $(X_{k-1})' = X_{k-2} + (\Delta_{k-1})' = X_{k-2} + \frac{1}{2}\Delta_{k-1}$

(c) $(\Delta_k)' = \Delta_k = -\frac{1}{2}\Delta_{k-1}$

(d) $(X_k)' = (X_{k-1})' + (\Delta_k)' = X_{k-2}$

(e) $(e_k)' = -e_k$

Step 7: Update k. That is, set $e_{k-3} = e_{k-2}$; $e_{k-2} = e_{k-1}$; if step 6 is executed $e_{k-1} = (e_k)'$, otherwise $e_{k-1} = e_k$, etc.

IV. HARDWARE IMPLEMENTATION OF THE OVER-SHOOT SUPPRESSION ALGORITHM

The implementation of the above overshoot suppression algorithm requires the addition of very little hardware to the Digital Song Adaptive Delta Modulator. This can be seen by considering the schematic representation of the delta modulator CODEC (Coder-Decoder- Combination) with overshoot suppression shown in Fig.2. Note that the extra components needed to implement the suppression scheme appear in branches that are drawn with dashed lines. Of these, the only major devices are the delay elements D5,D6, and the adder A4. However, since the adders A1,A2, and A3 are really one time-shared adder, we can easily time-share adder A4 also. The remaining extra elements are only a few gates needed for decision, switching, and timing purposes. Note that the execution of step 6(a) of the algorithm, $(\Delta_{k-1})' = \frac{1}{2}\Delta_{k-1}$, need not require the addition of any explicit hardware. We merely read into adder A4 the contents of the Δ_{k-1} register shifted by one bit, thereby resulting in a division by two.

It is difficult to discern the operation of the circuit by merely examining the schematic diagram in Fig.2 because the sequential order of operations is not not specified in the diagram. However, the actual operation is made clear by considering Fig.2 in conjunction with the seven steps of the overshoot suppression algorithm.

The additional steps of the algorithm place an added requirement on the logic speed. After the completion of a normal cycle of the delta modulator, extra time is needed to perform two more additions and the various logic operations needed to rearrange the internal values. No problem will arise if this can be done in one sampling period. If it cannot, then either the sampling rate must be decreased, or logic components capable of higher switching speed must be used.

At this point, it should be pointed out that a significant simplification is possible in the encoder implementation. Namely, Step 6(b) and hence Step 6(a) do not have to be explicitly executed in the encoder because $(X_{K-1})'$ is not really needed to compute $(X_K)'$ in Step 6(d), i.e. $(X_k)'$ is simply replaced by X_{k-2}, which is available in the memory. Furthermore, it is easy to snow that $(X_{k-1})'$ is not used in later cycles due to the fact that once an overshoot is suppressed at k-1, the earliest future time for implementing the algorithm is at k+2. By this time X_{k-1} is clocked out of the memory. In terms of hardware savings in the encoder, this eliminates the gates needed to produce $(\Delta_{k-1})' = \frac{1}{2}\Delta_{k-1}$, as well as A4 and A2 to carry out Steps 6(b) and 6(d) respectively. These simplifications are not possible in the decoder because its output with overshoot suppression, has to be taken from X_{k-1} rather than from X_k. Note that if the output was taken from X_k, then the overshoot suppression produced by going back in time and reducing X_{k-1} would not be evident in the output $\hat{S}(t)$.

V. COMPUTER SIMULATIONS

The Digital Song Delta Modulator, with and without overshoot suppression, was simulated on a PDP-8 computer. The minimum step size used (Δ_0) was normalized to unity. The dynamic range was 0 to 1024 Δ_0. This corresponds to a ten bit internal arithmetic in an actual hardware implementation.

The responses of the delta modulator to step functions of different amplitudes, with and without overshoot suppression, appear in Fig. 6. Figs. 6(a), and 6(b) exhibit large overshoots and sustained oscillations. They correspond to the sequence $e_{k-3} = e_{k-2} = 1$, $e_{k-1} = e_k = -1$, where k-1 is the sampling time when overshoot occurs. Figs. 6(a'), and 6(b') are the same waveforms but with overshoot suppression. As an example, compare Figs. 6(a) and 6(a'). Here the maximum peak-to-peak oscillations are reduced from $22\Delta_0$ to $9\Delta_0$. Similar observations can be made for Figs. 6(b) and 6(b'). Furthermore, here the settling time to the steady state is reduced from six to three sampling periods.

While Fig. 6 gives a good indication of the general nature of the improvement, due to overshoot suppression, a more convincing illustration is depicted in Fig.7 where the discontinuities are much larger. Note that the apparent slow rise times in Fig.7 are due to the compression produced by a scaling factor of 0.1 used in the plotting. In reality Fig.7 rises over a range of $5000\Delta_0$ in only 13 sampling periods. To achieve the same amplitude, a non-adaptive delta modulator would require 500 sampling periods.

Briefly, the salient features of the response are as follows: The rise time to reach a given level is the same with or without overshoot suppression. Overshoots are suppressed by a minimum of 50%. Recovery times following overshoots are significantly reduced as seen in Fig.7b. The data plotted in Fig. 7 is given in Table 1 for quantitative comparisons. The peak-to-peak amplitude of the steady state response is three times the minimum step size for either scheme. The period of steady state oscillations is 4 sampling periods without overshoot suppression, and 8 sampling periods with overshoot suppression. In either case, the peak-to-peak steady state oscillation amplitudes are smaller than a grey level in the picture waveform. Thus constant shade regions will not suffer significant degradation.

This is based on the assumption of quantizing a pixel into 64 grey levels with a total dynamic range of 1024Δ

VI. CONCLUSIONS

An overshoot suppression algorithm has been proposed and verified by computer simulation. It has been shown that the scheme significantly improves the transient behavior of video waveforms transmitted using delta modulation techniques.

The main advantages of the proposed algorithm are:

(a) It can be easily utilized in optimal digital delta modulators that can be described by a closed-form mathematical formulation and in particular in the Adaptive Song Delta Modulator.

(b) The scheme has rather modest requirements for hardware implementation..

(c) It allows for flexible trade-off between slope-overload and overshoot noise.

Therefore, the addition of the overshoot suppression algorithm significantly improves the performance of the digital delta modulator for picture transmission.

VII. REFERENCES

1. C.L. Song, J. Garodnick, and D.L.Schilling, "A Variable-Step-Size Robust Delta Modulator," IEEE Trans. Commun. Technol., Vol COM-19, pp. 1033-1044, Dec. 1971.

2. M.R. Winkler,"Pictorial Transmission with HIDM" in 1965 IEEE Int. Conv. Rec., pt. 1. pp.285-291.

3. M.Oliver, "An Adaptive Delta Modulator with Overshoot Suppression for Video Signals," IEEE Trans. Commun. Technol., Vol. COM-21, pp. 243-247, March 1973.

4. C.L. Song, J. Garodnick, D.L. Schilling, "An Adaptive Delta Modulator for Speech and Video Processing," Proceedings of the 1972 IEEE International Conference on Commun., Philadelphia, Pa., pp.21.30- 21.31, June, 1972.

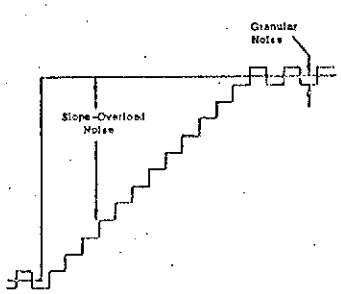


Fig. 1(a) Linear Delta Modulator step response exhibiting Slope-Overload-Noise and Granular Noise.

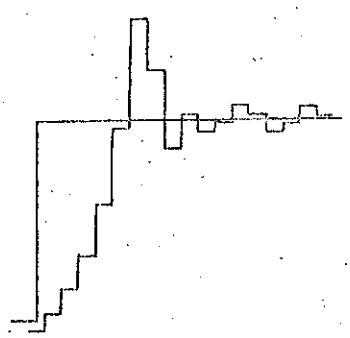


Fig. 1(b) Adaptive Delta Modulator step response exhibiting overshoots and oscillations.

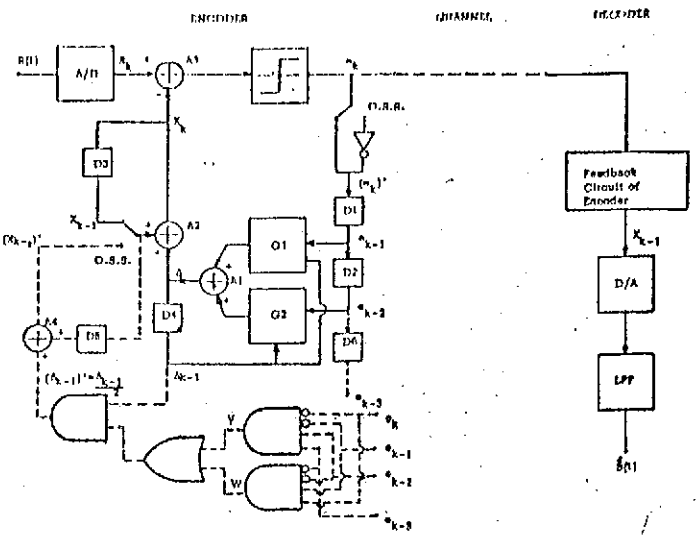


Fig. 2 Song-Adaptive-Delta Modulator with provisions for Overshoot Suppression. (Dashed branches are for Overshoot Suppression.)

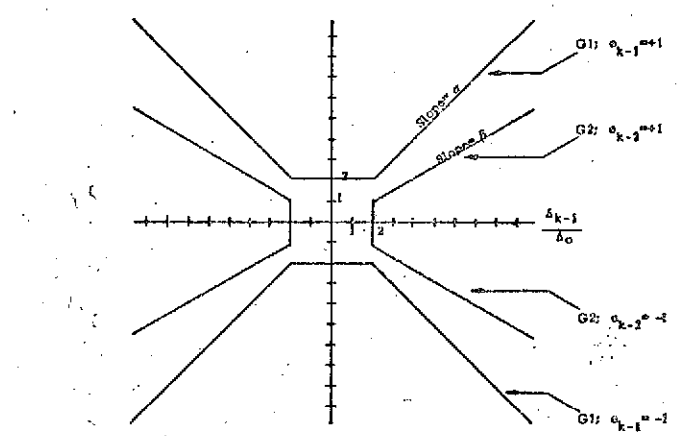


Fig. 3 Normalized G1, G2 Characteristics. (Δ_o = Minimum Step Size.)

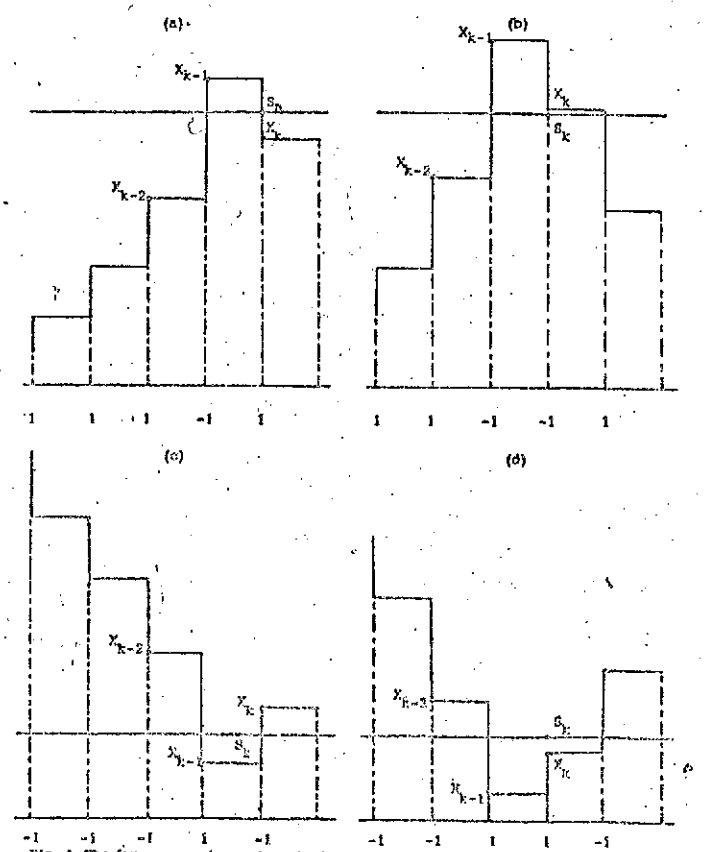


Fig. 4 The four cases of overshoot in the step response of the Song-Adaptive-Delta Modulator.

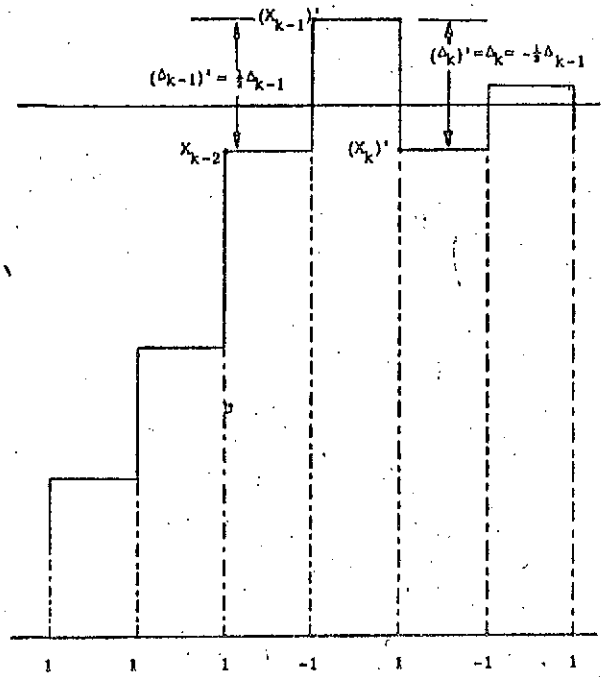


Fig. 5 Figure 4b after Overshoot Suppression.

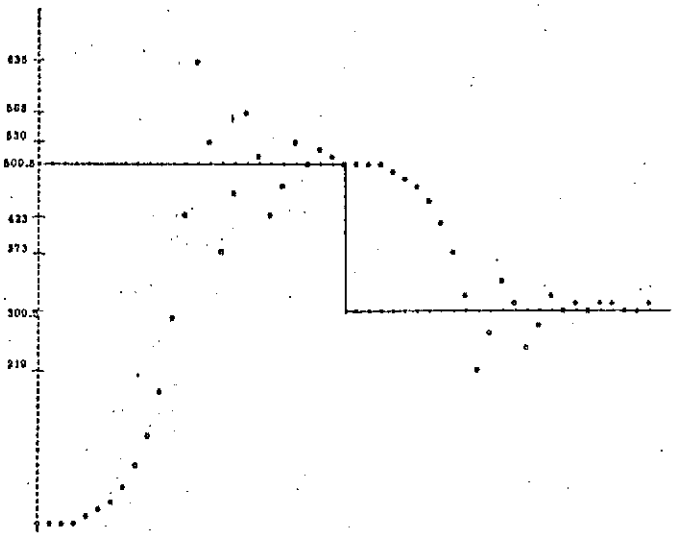


Fig. 7(a) The response of a Song-Adaptive Delta Modulator to abrupt level changes without Overshoot Suppression.

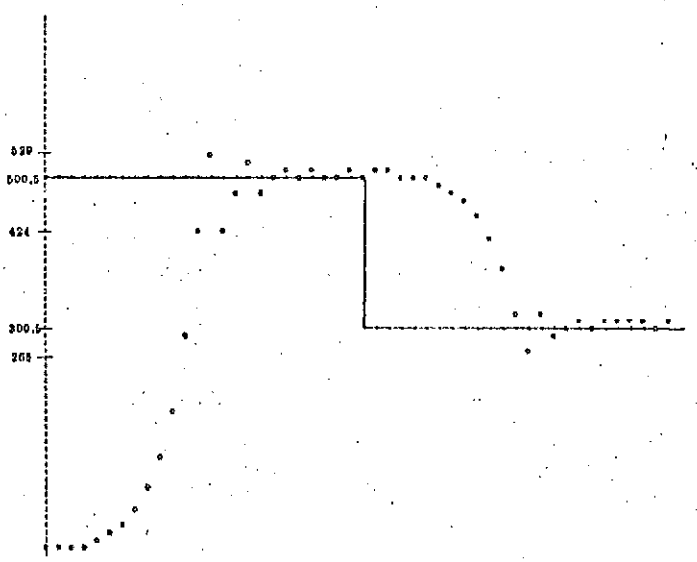


Fig. 7(b) The response of a Song-Adaptive Delta Modulator to abrupt level changes with Overshoot Suppression.

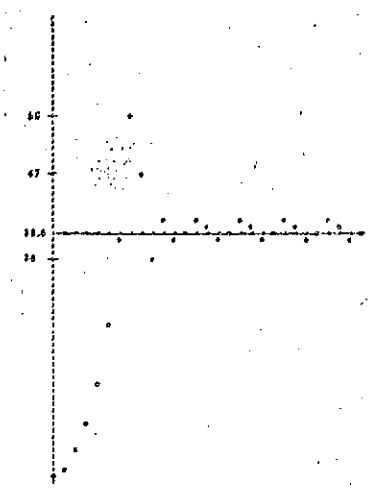


Fig. 6 (a) The step response of a Song-Adaptive Delta Modulator without Overshoot Suppression.

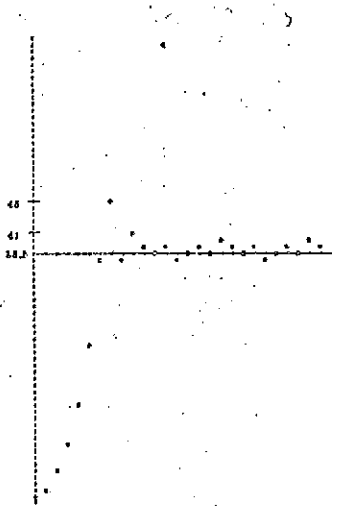


Fig. 6(a') The step response of a Song-Adaptive Delta Modulator with Overshoot Suppression.

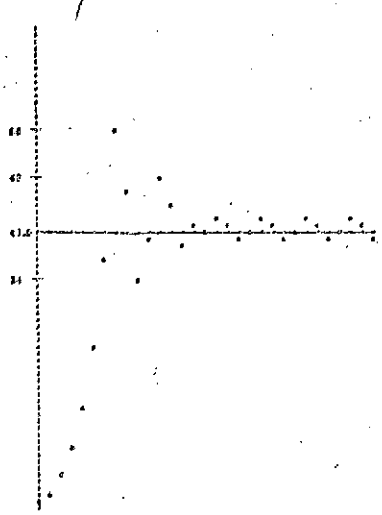


Fig. 6(b) The step response of a Song-Adaptive Delta Modulator without Overshoot Suppression.

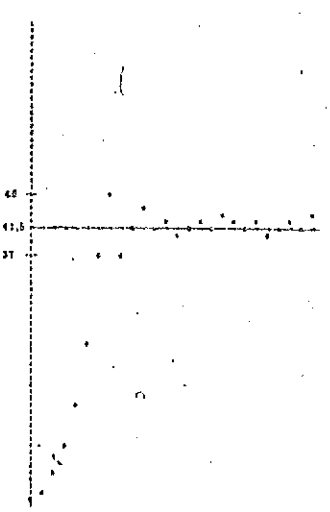


Fig. 6(b') The step response of a Song-Adaptive Delta Modulator with Overshoot Suppression.

K= 0.000	SK= 500.5	XK= 0.000	EKE= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKE= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKE= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKE= 1.000
K= 4.000	SK= 500.5	XK= 15.00	EKE= 1.000
K= 5.000	SK= 500.5	XK= 24.00	EKE= 1.000
K= 6.000	SK= 500.5	XK= 37.00	EKE= 1.000
K= 7.000	SK= 500.5	XK= 56.00	EKE= 1.000
K= 8.000	SK= 500.5	XK= 84.00	EKE= 1.000
K= 9.000	SK= 500.5	XK= 126.0	EKE= 1.000
K= 10.00	SK= 500.5	XK= 189.0	EKE= 1.000
K= 11.00	SK= 500.5	XK= 283.0	EKE= 1.000
K= 12.00	SK= 500.5	XK= 424.0	EKE= 1.000
K= 13.00	SK= 500.5	XK= 635.0	EKE=-1.000
K= 14.00	SK= 500.5	XK= 530.0	EKE=-1.000
K= 15.00	SK= 500.5	XK= 373.0	EKE= 1.000
K= 16.00	SK= 500.5	XK= 451.0	EKE= 1.000
K= 17.00	SK= 500.5	XK= 560.0	EKE=-1.000
K= 18.00	SK= 500.5	XK= 510.0	EKE=-1.000
K= 19.00	SK= 500.5	XK= 423.0	EKE= 1.000
K= 20.00	SK= 500.5	XK= 466.0	EKE= 1.000
K= 21.00	SK= 500.5	XK= 530.0	EKE=-1.000
K= 22.00	SK= 500.5	XK= 498.0	EKE= 1.000
K= 23.00	SK= 500.5	XK= 514.0	EKE=-1.000
K= 24.00	SK= 500.5	XK= 506.0	EKE=-1.000
K= 25.00	SK= 500.5	XK= 494.0	EKE= 1.000
K= 26.00	SK= 300.5	XK= 500.0	EKE=-1.000
K= 27.00	SK= 300.5	XK= 497.0	EKE=-1.000
K= 28.00	SK= 300.5	XK= 493.0	EKE=-1.000
K= 29.00	SK= 300.5	XK= 487.0	EKE=-1.000
K= 30.00	SK= 300.5	XK= 478.0	EKE=-1.000
K= 31.00	SK= 300.5	XK= 465.0	EKE=-1.000
K= 32.00	SK= 300.5	XK= 446.0	EKE=-1.000
K= 33.00	SK= 300.5	XK= 418.0	EKE=-1.000
K= 34.00	SK= 300.5	XK= 376.0	EKE=-1.000
K= 35.00	SK= 300.5	XK= 313.0	EKE=-1.000
K= 36.00	SK= 300.5	XK= 219.0	EKE= 1.000
K= 37.00	SK= 300.5	XK= 266.0	EKE= 1.000
K= 38.00	SK= 300.5	XK= 336.0	EKE=-1.000
K= 39.00	SK= 300.5	XK= 301.0	EKE=-1.000
K= 40.00	SK= 300.5	XK= 249.0	EKE= 1.000
K= 41.00	SK= 300.5	XK= 275.0	EKE= 1.000
K= 42.00	SK= 300.5	XK= 314.0	EKE=-1.000
K= 43.00	SK= 300.5	XK= 295.0	EKE= 1.000
K= 44.00	SK= 300.5	XK= 304.0	EKE=-1.000
K= 45.00	SK= 300.5	XK= 300.0	EKE= 1.000
K= 46.00	SK= 300.5	XK= 302.0	EKE=-1.000
K= 47.00	SK= 300.5	XK= 301.0	EKE=-1.000
K= 48.00	SK= 300.5	XK= 299.0	EKE= 1.000
K= 49.00	SK= 300.5	XK= 300.0	EKE= 1.000
K= 50.00	SK= 300.5	XK= 300.0	EKE=-1.000

Table 1(a) Table of values for Fig. 7(e)
K= K'th sampling instant.
SK = input signal at time K.
XK = input estimate at time K.
EKE = sign bit at time K.

K= 0.000	SK= 500.5	XK= 0.000	EKU= 1.000	EKT= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKU= 1.000	EKT= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKU= 1.000	EKT= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKU= 1.000	EKT= 1.000
K= 4.000	SK= 500.5	XK= 15.00	EKU= 1.000	EKT= 1.000
K= 5.000	SK= 500.5	XK= 24.00	EKU= 1.000	EKT= 1.000
K= 6.000	SK= 500.5	XK= 37.00	EKU= 1.000	EKT= 1.000
K= 7.000	SK= 500.5	XK= 56.00	EKU= 1.000	EKT= 1.000
K= 8.000	SK= 500.5	XK= 84.00	EKU= 1.000	EKT= 1.000
K= 9.000	SK= 500.5	XK= 126.0	EKU= 1.000	EKT= 1.000
K= 10.00	SK= 500.5	XK= 189.0	EKU= 1.000	EKT= 1.000
K= 11.00	SK= 500.5	XK= 283.0	EKU= 1.000	EKT= 1.000
K= 12.00	SK= 500.5	XK= 424.0	EKU= 1.000	EKT= 1.000
K= 13.00	SK= 500.5	XK= 529.0	EKU=-1.000	EKT=-1.000
K= 14.00	SK= 500.5	XK= 424.0	EKU= 1.000	EKT=-1.000
K= 15.00	SK= 500.5	XK= 476.0	EKU= 1.000	EKT= 1.000
K= 16.00	SK= 500.5	XK= 515.0	EKU=-1.000	EKT=-1.000
K= 17.00	SK= 500.5	XK= 476.0	EKU= 1.000	EKT=-1.000
K= 18.00	SK= 500.5	XK= 495.0	EKU= 1.000	EKT= 1.000
K= 19.00	SK= 500.5	XK= 509.0	EKU=-1.000	EKT=-1.000
K= 20.00	SK= 500.5	XK= 495.0	EKU= 1.000	EKT=-1.000
K= 21.00	SK= 500.5	XK= 502.0	EKU=-1.000	EKT=-1.000
K= 22.00	SK= 500.5	XK= 499.0	EKU= 1.000	EKT= 1.000
K= 23.00	SK= 500.5	XK= 500.0	EKU= 1.000	EKT= 1.000
K= 24.00	SK= 500.5	XK= 501.0	EKU=-1.000	EKT=-1.000
K= 25.00	SK= 500.5	XK= 500.0	EKU= 1.000	EKT=-1.000
K= 26.00	SK= 300.5	XK= 502.0	EKU=-1.000	EKT=-1.000
K= 27.00	SK= 300.5	XK= 501.0	EKU=-1.000	EKT=-1.000
K= 28.00	SK= 300.5	XK= 499.0	EKU=-1.000	EKT=-1.000
K= 29.00	SK= 300.5	XK= 496.0	EKU=-1.000	EKT=-1.000
K= 30.00	SK= 300.5	XK= 492.0	EKU=-1.000	EKT=-1.000
K= 31.00	SK= 300.5	XK= 486.0	EKU=-1.000	EKT=-1.000
K= 32.00	SK= 300.5	XK= 477.0	EKU=-1.000	EKT=-1.000
K= 33.00	SK= 300.5	XK= 464.0	EKU=-1.000	EKT=-1.000
K= 34.00	SK= 300.5	XK= 445.0	EKU=-1.000	EKT=-1.000
K= 35.00	SK= 300.5	XK= 417.0	EKU=-1.000	EKT=-1.000
K= 36.00	SK= 300.5	XK= 375.0	EKU=-1.000	EKT=-1.000
K= 37.00	SK= 300.5	XK= 312.0	EKU=-1.000	EKT=-1.000
K= 38.00	SK= 300.5	XK= 265.0	EKU= 1.000	EKT= 1.000
K= 39.00	SK= 300.5	XK= 312.0	EKU=-1.000	EKT= 1.000
K= 40.00	SK= 300.5	XK= 289.0	EKU= 1.000	EKT= 1.000
K= 41.00	SK= 300.5	XK= 300.0	EKU= 1.000	EKT= 1.000
K= 42.00	SK= 300.5	XK= 306.0	EKU=-1.000	EKT=-1.000
K= 43.00	SK= 300.5	XK= 300.0	EKU= 1.000	EKT=-1.000
K= 44.00	SK= 300.5	XK= 3040.0	EKU=-1.000	EKT=-1.000
K= 45.00	SK= 300.5	XK= 302.0	EKU=-1.000	EKT=-1.000
K= 46.00	SK= 300.5	XK= 301.0	EKU= 1.000	EKT= 1.000
K= 47.00	SK= 300.5	XK= 302.0	EKU=-1.000	EKT= 1.000
K= 48.00	SK= 300.5	XK= 300.0	EKU= 1.000	EKT= 1.000
K= 49.00	SK= 300.5	XK= 301.0	EKU=-1.000	EKT=-1.000

Table 1 (b) Table of values for Fig. 7(b)
EKU = sign bit used inside encoder and decoder.
EKT = sign bit transmitted from encoder to decoder.